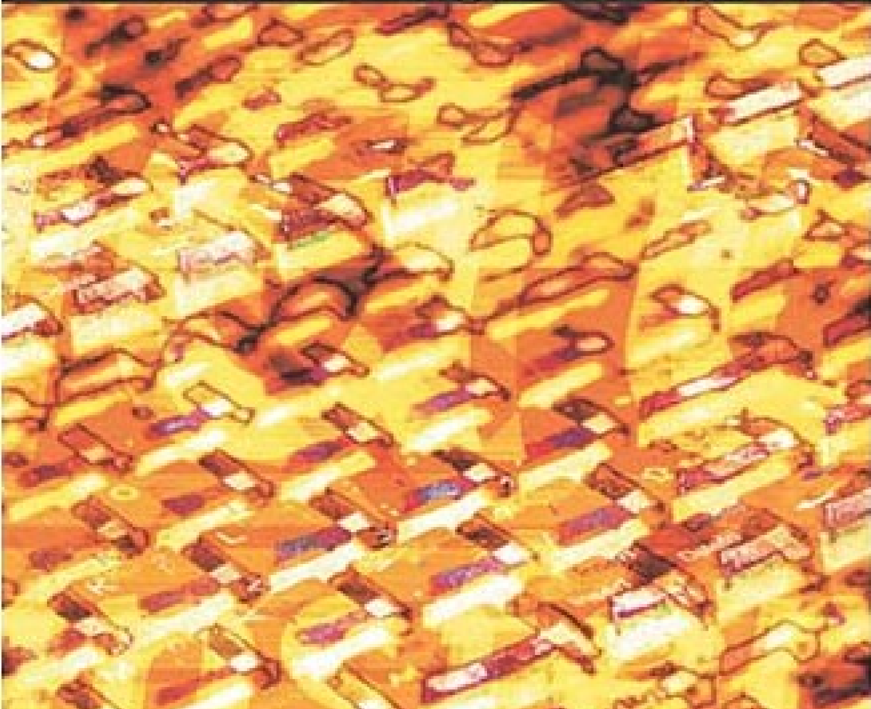
VISIONS OF MIND

Architectures for Cognition and Affect



DARRYL N. DAVIS

Visions of Mind:
Architectures for Cognition and Affect

Darryl N. Davis
University of Hull, Kingston-upon-Hull, UK

INFOSCI Information Science Publishing
Hershey • London • Melbourne • Singapore

Acquisition Editor:	Mehdi Khosrow-Pour
Senior Managing Editor:	Jan Travers
Managing Editor:	Amanda Appicello
Development Editor:	Michele Rossi
Copy Editor:	Lori Eby
Typesetter:	Jennifer Wetzel
Cover Design:	Lisa Tosheff
Printed at:	Yurchak Printing Inc.

Published in the United States of America by
Information Science Publishing (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue, Suite 200
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.idea-group.com

and in the United Kingdom by
Information Science Publishing (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 3313
Web site: http://www.eurospan.co.uk

Library of Congress Cataloging-in-Publication Data

Visions of mind : architectures for cognition and affect / Daryl N. Davis, editor.
p. cm.
Includes bibliographical references and index.
ISBN 1-59140-482-7 (hard) -- ISBN 1-59140-483-5 (pbk.) -- ISBN 1-59140-484-3 (ebook)
1. Artificial intelligence. 2. Cognitive science. I. Davis, Daryl N., 1955-
Q335.V57 2004
006.3--dc22

2004022148

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

All work contributed to this book is new, previously-unpublished material. The view expressed in this book are those of the authors, but not necessarily of the publisher.

Visions of Mind:
Architectures for Cognition and Affect

Table of Contents

Preface:
Architectures for Cognition and Affect

Introduction

In Spring 2000, the annual symposium of *The Society for the Study of Artificial Intelligence and the Simulation of Behaviour* (AISB) was held at the University of Birmingham. One of the principal workshops ran under the title "How to Design a Functioning Mind." This was organized by Professor Aaron Sloman and attracted a set of international speakers. The following section summarizes the aim, objective, and purpose of that symposium.

This text has similar objectives. It draws on those authors present at the workshop plus others to provide updates on their perspectives for this collected text. The main objective of this collection, nearly 50 years from the (in)famous Dartmouth Conference (McCarthy et al., 1956) where the term "Artificial Intelligence" was first used, is to present an overview of where the research area of artificial minds is at the start of the 21st century. As such, it draws on current and prospective works from pivotal researchers in the area. It includes perspectives from philosophy, psychology, social studies, cognitive science, and artificial intelligence. Rather than present a unified model of mind, it proffers a diverse collection that mirrors the academic discipline. The objectives and mission of the text are, in part, a rephrasing of those associated with the original workshop.

This text brings together research from academics in Europe and the United States interested in building bridges between various kinds of partial studies, with the long-term goal of understanding, at least in principle, how to build a complete mind. This topic is important to artificial intelligence and cognitive science, and also to the academic disciplines that they draw on and feed from, for instance, philosophy, computer science, and psychology. The collection of chapters follows in the tradition of *Computers and Thought* (Feigenbaum & Feldman, 1963), *Mind Design* (Haugeland, 1981), *Architectures for Intelligence* (VanLehn, 1988), and *Android Epistemology* (Ford, Glymour, & Hayes, 1989).

Symposium: How to Design a Functioning Mind

The objective of the two-day symposium was to adopt a multidisciplinary approach to the long-term problem of designing a human-like mind, whether for the scientific purpose of understanding human minds or for some engineering purpose. The following summary of the purpose of the symposium is an edited and abridged version of the original call, which can be found on the Internet (Sloman, 2000).

Purpose of the Symposium

The symposium was intended to bring together people interested in building bridges between various kinds of partial studies (in artificial intelligence, biology, computer science, engineering, ethology, philosophy, and psychology), with the long-term goal of understanding, at least in principle, how to build a complete mind.

Researchers in any discipline were invited to address this issue, whether in a speculative fashion or by reporting firm results that directly contribute to the long-term task. Examples of topics proposed included the following:

- Architectures to accommodate multiple aspects of human mental functioning
- Analyses of requirements for such architectures
- Critiques of existing architectures on the bases of their functional limitations or inconsistent empirical evidence
- Discussions of how important aspects of human minds might have evolved
- Analyses of the problems of designing an adult mind versus designing an infant mind that develops into an adult mind
- Comparisons between capabilities of different animals, which provides evidence for architectural differences
- Overviews of major results in neuroscience that have implications for the virtual machine architecture of a mind (for example, evidence from brain-damaged patients indicating what sorts of separable functional modules exist)
- Philosophical posters presenting familiar arguments to prove that the task is impossible were not particularly welcome, whereas philosophical arguments highlighting some of the difficulties to be overcome were.

Setting the Scene

This text is concerned with the study of questions such as the following:

- What is mind?
- What are theories of mind?
- What are computationally plausible theories of mind?
- What are computationally plausible designs based on these theories of mind?
- What are computationally plausible architectures and systems to support theories of mind?
- What are computationally implemented architectures and systems based on theories of mind?
- What kinds of tools can we use in producing and implementing such designs?
- How do we know when we are successful in producing artificial minds?
- Why do we want to produce artificial or synthetic minds?

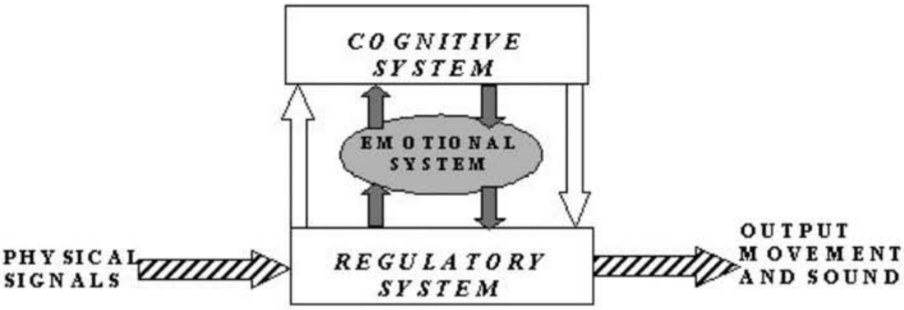
Such questions have been addressed throughout the history of western civilization, from the philosophers of ancient Greece (see Popkin & Stroll, 1993, and Bechtel, 1988, for an overview), through the Age of Enlightenment, for example, Descartes (Bechtel, 1988), the industrial revolution, and Babbage's designs for the Difference Engine (Swage, 2001), to the advent of the computer (von Neumann, 1963; Turing, 1950) and, since its inception as an academic field in its own right, in artificial intelligence (McCarthy et al., 1956).

Just over 20 years ago, Donald Norman (1980) set an agenda of important issues for cognitive science: belief systems, consciousness, development, emotion, interaction, language, learning, memory, perception, performance, skill, and thought. This nonexhaustive list of topics has played an important role in determining where researchers have focused their work.

Such study is arguably the core of the field of artificial intelligence (Barr & Feigenbaum, 1981; Franklin, 1995; Minsky, 1987; Nilsson, 1998; Russell & Norvig, 2003; Sharples et al., 1989; Winston, 1992) and cognitive science (Bechtel, 1988; Bechtel & Abrahamsen, 2002; Simon, 1979; Wilson & Keil, 1999; Winograd & Flores, 1986).

An important aspect of the agenda set by Norman is the use of an architectural perspective. One such example architecture is given in Figure 1. Five interacting processes are identified: the reception of incoming signals (perception), the

Figure 1. One architecture for an artificial mind (based on Norman, 1980)



generation of output (action selection), a reactive or regulatory system, a deliberative or cognitive system, and an emotional or affect system. Norman suggested that this is the type of architecture needed to address the 12 topics included in his agenda for cognitive science. The 15 chapters in this text address these and many related issues.

Chapter Overview

In Chapter 1, Andrew Adamatzky portrays the mind as an imaginary chemical reactor, where discrete entities of emotions and beliefs diffuse and react as molecules. He presents two models of mind using the computational chemistry metaphor: doxastic solution where quasi-chemical species represent knowledge, ignorance, delusion, doubt, and misbelief; and affective solution, where reaction mixtures include happiness, anger, confusion, and fear. Using numerical and cellular-automaton techniques, he presents a rich spectrum of nontrivial phenomena in the spatiotemporal dynamic of the affective and doxastic mixtures. This paradigm of nonlinear medium-based mind is to be used in future studies in developing intelligent robotic systems, designs of artificial organic creatures with liquid brains, and the diffusive intelligence of agent collectives.

In Chapter 2, Michel Aubé proposes a model of emotions relying upon an analysis of the requirements that are to be met by individuals of nurturing species, so as to adapt themselves to their social environments. It closely reflects the structures of other motivational systems which consist of control structures dedicated to the management of resources critical for survival. The particular resources emotional systems seem to handle have to do with social bonding and collaborative behaviors. These are referred to as second-order resources. They are made available to other agents, and they are captured in the model through the concept of commitments. Emotions thus appear as computational control systems that handle the variation of commitments lying at the root of interactive and collaborative behaviors. Some critical consequences of the model for the implementation of emotions in artificial systems are drawn at the end of the chapter.

In Chapter 3, John Barnden speculatively addresses the nature and effects of metaphorical views that a mind can intermittently use in thinking about itself and other minds, such as the view of mind as a physical space in which ideas have physical locations. Although such views are subjective, the chapter argues that they are nevertheless part of the real nature of the conscious and unconscious mind. In particular, it is conjectured that if a mind entertains a particular (metaphorical) view at a given time, then this activity could of itself cause that mind to become more similar in the short term to how it is portrayed by the view. Hence, the views are, to an extent, self-fulfilling prophecies. In these ways, metaphorical self-reflection, even when distorting and inaccurate, is speculatively an important aspect of the true nature of mind. The chapter also outlines a theoretical approach and related implemented system (ATT-Meta) that were designed for the understanding of metaphorical discourse but that have principles that could be at the core of metaphorical self-reflection in people or future artificial agents.

In Chapter 4, Joanna Bryson presents an analysis of the modularity of mind. Many architectures of mind assume some form of modularity, but what is meant by the term "module"? This chapter creates a framework for understanding current modularity research in three subdisciplines of cognitive science — psychology, artificial intelligence, and neuroscience. The framework starts from the distinction between horizontal modules that support all expressed behaviors versus vertical modules that support individual domain-specific capacities. The framework is used to discuss innateness, automaticity, compositionality, representations, massive modularity, behavior-based and multiagent artificial intelligence systems, and correspondence to physiological neurosystems. There is also a brief discussion of the relevance of modularity to conscious experience.

In Chapter 5, William Clocksin explores issues in memory and affect in connection with possible architectures for artificial cognition. Because memory and emotion are evolutionarily and developmentally rooted in social interdependence, a new understanding of these issues is necessary for the principled design of true intelligent systems. Emotion is not treated as an optional extra or as a brief episode of feeling but as the underlying substrate enabling the formation of social relationships essential for the construction of cognition. Memory is not treated as the storage and retrieval of immutable data but as a continuous process contingent on experience and never fully fixed or immutable. Three converging areas of research are identified that hold some promise for further research: social constructionism, reconsolidation memory theory, and memory models based on the nonlinear dynamics of unstable periodic orbits. He argues that the combination of these ideas offers a potentially more substantive approach to understanding the cognitive architecture.

In Chapter 6, Bruce Edmonds describes free will in terms of the useful properties that it could confer, explaining why it might have been selected over the course of evolution. These properties are exterior unpredictability, interior rationality, and social accountability. A process is described that might bring it about when deployed in a suitable social context. It is suggested that this process could be of an evolutionary nature, that free will might "evolve" in the brain during development. This mental evolution effectively separates the internal and external contexts, while retaining the coherency between individuals' public accounts of their actions. This is supported by the properties of evolutionary algorithms and possesses the three desired properties. Some objections to the possibility of free will are dealt with by pointing out the *prima facie* evidence and showing how an assumption that everything must be either deterministic or random can result from an unsupported assumption of universalism.

In Chapter 7, John Fox presents argumentation about the concept of mind. The idea of "mind" did not spring fully formed into human consciousness. On the contrary, it has been articulated slowly through the millennia, drawing upon countless metaphors and images in different cultures at different times. In the last 50 years, the concepts of conventional science and technology provided the primary images that we employ in discussing mental processes, though there are presently many competing perspectives. Each of these images is incomplete when it comes to explaining mental phenomena, and many are inconsistent. In this chapter, a few of the prominent perspectives that influenced cognitive science in the last half century or so, from information processing psychology to artificial intelligence, are reviewed. The conclusion is that a unified theory of mind will need insights from multiple viewpoints. The challenge to the field is to avoid disputes over different positions and look for ways to bring them together.

In Chapter 8, Stan Franklin describes an architecture for an autonomous software agent designed to model a broad spectrum of human cognitive and affective functioning. In addition to featuring "consciousness," the architecture accommodates perception, several forms of memory, emotions, action-selection, deliberation, ersatz language generation, several forms of learning, and metacognition. One such software agent, IDA, embodying much of this architecture, is up and running. IDA's "consciousness" module is based on global workspace theory, allowing it to select relevant resources with which to deal flexibly with exogenous and endogenous stimuli. Within this architecture, emotions implement IDA's drives, her primary motivations. Offering one possible architecture for a fully functioning artificial mind, IDA constitutes an early attempt at the exploration of design space and niche space. The design of the IDA architecture spawns hypotheses concerning human cognition and affect that can serve to guide the research of cognitive scientists and neuroscientists.

In Chapter 9, David Glasspool discusses how clues to the way behaviour is integrated and controlled in the human mind emerged from cognitive psychology and neuroscience. The emerging picture mirrors solutions (driven primarily by engineering concerns) to similar problems in the rather different domains of mobile robotics and intelligent agents in artificial intelligence. Details are presented about the relationship between a psychological theory of willed and automatic control of behaviour, the Norman and Shallice framework, and three types of engineering-based theory in artificial intelligence. As well as being a promising basis for a large-scale model of cognition, the Norman and Shallice framework presents an interesting example of apparent theoretical convergence between artificial intelligence and empirical psychology, and of the ways in which theoretical work in both fields can benefit from interaction between them.

In Chapter 10, Fernand Gobet and Peter C.R. Lane provide an introduction to the CHREST architecture of cognition and show how this architecture can help develop a full theory of mind. After describing the main components and mechanisms of the architecture, they discuss several domains in which it was already successfully applied, such as in the psychology of expert behaviour, the acquisition of language by children, and the learning of multiple representations in physics. The highlighted characteristics of CHREST that enable it to account for empirical data include self-organisation, an emphasis on cognitive limitations, the presence of a perception-learning cycle, and the use of naturalistic data as input for learning. They argue that some of these characteristics can help shed light on the hard questions facing theorists developing a full theory of mind, such as intuition, the acquisition and use of concepts, the link between cognition and emotions, and the role of embodiment.

In Chapter 11, Elizabeth Gordon and Brian Logan address the key problem for agents of responding in a timely and appropriate way to multiple, often conflicting goals in a complex, dynamic environment. They propose a novel goal-processing architecture that allows an agent to arbitrate between multiple conflicting goals. Building on the teleo-reactive programming framework originally developed in robotics, they introduce the notion of a *resource*, which represents a condition that must be true for the safe concurrent execution of a durative action. They briefly outline a goal arbitration architecture for teleo-reactive programs with resources, which allows an agent to respond flexibly to multiple competing goals with conflicting resource requirements.

In Chapter 12, Pentti Haikonen considers the following fundamental issues of artificial minds and conscious machines: the representation and symbolic processing of information with meaning and significance in the human sense; the perception process; a neural cognitive architecture; system reactions and emotions; consciousness in the machine; and artificial minds as a content-level phenomenon. Solutions are proposed for related problems, and a cognitive machine is outlined. An artificial mind within this machine that eventually controls the machine, is seen to arise via learning and experience as higher-level content is constructed.

In Chapter 13, Colin Johnson considers how new and emerging computational models relate to the arguments surrounding dualism and embodiment. In recent years, the idea that somatic processes are intimately involved in actions traditionally considered to be purely mental has come to the fore. These arguments have revolved around the concept of *somatic markers*, i.e., bodily states generated by the mind and then reperceived and acted upon. In this chapter, the somatic marker hypothesis and related ideas from the point of view of *postclassical computation*, i.e., the view that computing can be seen as a property of things-in-the-world rather than of an abstract class of mathematical machines, are considered. From this perspective, a number of ideas are discussed: the idea of somatic markers extending into the environment, an analogy with hardware interlocks in complex computer-driven systems, and connections with the idea of "just-do-it" computation.

In Chapter 14, Matthias Scheutz introduces an *architecture framework* called *APOC* (for "*Activating-Processing-Observing-Components*") for the analysis, evaluation, and design of complex agents. APOC provides a unified framework for the specification of agent architectures at different levels of abstraction. As such, it permits intermediary levels of architectural specification between high-level functional descriptions and low-level mechanistic descriptions that can be used to systematically connect these two levels.

In Chapter 15, Push Singh and Marvin Minsky consider how to build systems with "common sense," the thinking skills that every ordinary person takes for granted (?). To build systems as resourceful and adaptive as people, we must develop cognitive architectures that support great procedural and representational diversity. No single technique is powerful enough to deal with the broad range of domains every ordinary person can understand — even as children, we can effortlessly think about complex problems involving temporal, spatial, physical, bodily, psychological, and social dimensions. In their chapter, they describe a multiagent cognitive architecture that aims for such flexibility. Rather than seek a best way to organize agents, our architecture supports multiple "ways to think," each a different architectural configuration of agents. Each agent may use a different way to represent and reason with knowledge, and there are special "panalogy" mechanisms that link agents that represent similar ideas in different ways. At the highest level, the architecture is arranged as a matrix of agents: Vertically the architecture divides into a tower of reflection, including the reactive, deliberative, reflective, self-reflective, and self-conscious levels; horizontally the architecture divides along "mental realms," including the temporal, spatial, physical, bodily, social, and psychological realms.

References

AISB. (n.d.). *The society for the study of artificial intelligence and the simulation of behaviour*. Retrieved from the World Wide Web: *http://www.aisb.org.uk/*

Barr, A., & Feigenbaum, E. A. (1981). *The handbook of artificial intelligence* (vol. 1). San Francisco: Morgan Kaufmann.

Bechtel, W. (1988). *Philosophy of mind: An overview of cognitive science*. Mahweh, NJ: Lawrence Erlbaum Associates.

Bechtel, W., & Abrahamsen, A. (2002). *Connectionism and the mind* (2nd ed.). Oxford: Blackwell Scientific.

Feigenbaum, E. A., & Feldman, J. (Eds.). (1963). *Computers and thought*. New York: McGraw-Hill.

Ford, K. M., Glymour, C., & Hayes, P. J. (Eds.). (1989). *Android epistemology*. Cambridge, MA: MIT Press.

Franklin, S. P. (1995). *Artificial minds*. Cambridge, MA: MIT Press.

Haugeland, J. (ed.). (1981). *Mind design*. Cambridge, MA: MIT Press.

McCarthy, J., Minsky, M. L., Rochester, N., & Shannon, C. E. (1956). *A proposal for the Dartmouth summer research project on artificial intelligence*. Retrieved from the World Wide Web: *http://www-formal.stanford.edu/jmc/history/dartmouth/dartmouth.html*

Minsky, M. L. (1987). *The society of mind*. London: William Heinemann Ltd.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Nilsson, N. J. (1998). *Artificial intelligence: A new synthesis*. San Francisco: Morgan Kaufmann.

Norman, D. A. (1980). Twelve issues for cognitive science. *Cognitive Science, 4*, 1-33.

Popkin, R. H., & Stroll, A. (1993). *Philosophy* (3rd ed.). London: Reed Elsevier.

Russell, S., & Norvig, P. (2003). *Artificial intelligence: A modern approach* (2nd ed.). New York: Prentice Hall.

Sharples, M., Hogg, D., Hutchinson, C., Torrance, S., & Young, D. (1980). *Computers and thought*. Cambridge, MA: MIT Press.

Simon, H. A. (1979). *Models of thought*. New Haven, CT: Yale University Press.

Sloman, A. (2000). *AISB 2000: The cognition and affect project*. University of Birmingham, UK. Retrieved from the World Wide Web: *http://www.cs.bham.ac.uk/research/cogaff/cogaff.html*

Swage, D. (2001). *The cogwheel brain: Charles Babbage and the quest to build the first computer*. Kent, UK: Abacus Press.

Turing, A. (1950). Computing machinery and intelligence. *Mind, 59*, 433-460.

VanLehn, K. (Ed.). (1988). *Architectures for intelligence*. Mahweh, NJ: Lawrence Erlbaum Associates.

von Newmann, J. (1963, originally published in 1937). General and logical theory of automata. In A. H. Taub (Ed.), *J. von Newmann, Collected works*. London: Pegamon/Elsevier.

Wilson, R. A., & Keil, F. C. (Eds.). (1999). *The MIT Encyclopedia of the cognitive sciences*. Cambridge, MA: MIT Press.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition*. Reading, MA: Addison-Wesley.

Winston, P. H. (1992). *Artificial intelligence* (3rd ed.). Reading, MA: Addison-Wesley.

Acknowledgments

Foremost on the list of acknowledgments, with their endless stream of helpful questions, are the chapter authors who responded so well to the editor's polite tyranny. Thanks also go to Keith Oatley, who on determining he would be unable to contribute directly, suggested contacting Michel Aubé for a psychological perspective.

Thanks are due to the publishing team at Idea Group Inc. In particular to Jan Travers and Michele Rossi, who responded to all my e-mail questions on the nature and format of the book once the project was underway, and to Mehdi Khosrow-Pour, who enthusiastically took the project on board when I was seeking an outlet.

Thanks are due to my department, and in particular, Derek Wills, for granting me the latitude to pursue this project, and those staff members involved in the endless discussions aspects of this text have provoked. Finally, thanks to Aaron Sloman who allowed me to take this little baby off his hands.

Darryl N. Davis, May 2004

Chapter 1

Parachemistry of Mind: Case Studies of Doxastic and Affective Mixtures

Andy Adamatzky
University of the West of England, UK

Abstract

We portray mind as an imaginary chemical reactor, where discrete entities of emotions and beliefs diffuse and react as molecules. We discuss two models of mind: a doxastic solution where quasi-chemical species represent knowledge, ignorance, delusion, doubt, and misbelief; and an affective solution, where reaction mixtures include happiness, anger, confusion, and fear. Using numerical and cellular-automaton techniques, we demonstrate a rich spectrum of nontrivial phenomena in the spatiotemporal dynamic of the affective and doxastic mixtures. This paradigm of nonlinear medium-based mind will be used in future studies in developing intelligent robotic systems, designs of artificial organic creatures with liquid brains, and diffusive intelligence of agent collectives.

Introduction

In this chapter, we crossbreed distributed concepts of mind (Minsky, 1988), theories of contagion of knowledge and emotions (Dawkins, 1976; Hatfield et al., 1994), swarm intelligence and collective pattern formation (Bonabeau et al., 1999), artificial chemistry (Dittrich et al., 2001), unconventional computing (Adamatzky, 2001), physical models of consciousness (Hameroff, 2001), and emotional intelligence (Goleman, 1996) to develop a nonlinear media-based representation of mind, thus building a bridge between conventional understanding of mind, dynamical psychology (Abraham & Gilgen, 1995), and sociophysics (Stauffer, 2003).

We model mind as a massive pool of simply interacting primitive entities, with spatiotemporal dynamics that reflect inner processes of interaction of affective and cognitive systems. We design, experiment with, and study *chemical*-like models of collectives of simply interacting affective and cognitive entities. Two models are considered here. An affective mixture, which represents happiness, anger, confusion, and fear as diffusing and reacting chemical species; and a doxastic mixture, where the reagents are knowledge, ignorance, delusion, doubt, and misbelief. These two types of mixtures are isolated from each other; no interaction between affective and doxastic components is discussed in the chapter. This isolation gives us an opportunity to look at a mind dissected into emotional and belief constituents. We derive several scenarios of quasi-chemical doxastic and affective reactions that exhibit nontrivial behaviour. We study global dynamic reagent concentrations in stirred reactors, where entities can interact at any distance; and spatial dynamics in thin-layer reactors, where entities almost do not move, and thus form a regular structure of locally interacting sites. We demonstrate that, depending on the particular architecture of reactions in doxastic and affective solutions, complex regimes of spatiotemporal dynamics can be observed, namely travelling waves, breathing domains, ordered patterns, and chaotic processes. The findings are substantiated by psychological correlates to indicate possible ways of mastering mind via controlling reaction mechanisms between doxastic and affective species.

Nonlinear Minds

Mathematical, computing, and physical sciences based approaches to the study of social and mental dynamics has a long, successful history. It probably started as early as 1940, when Jerome Lettvin and Walter Pits (Lettvin & Pits, 1943) designed a differential equation model of mental disorders. Their model represented a mental state as a tuple of two variables: intensity of emotions and intensity of activity. Mental dynamics was seen as a set of trajectories in phase spaces, for example, an increase of emotion intensity reflects a transition from impassivity to normal feeling, then to strong emotions, and finally to an abnormal excitement. A range of prospective approaches to mathematical modelling of mental processes was also highlighted by Richard Bellman in his invited address to the 1967 IEEE meeting (Bellman, 1967). Several decades later, these ideas, enriched by Robert Plutchik's three-dimensional structure of emotions (Plutchik, 2000), contributed to the emergence of a novel field of dynamical, or nonlinear, psychology (Lewis & Granic, 2000; Lewis & Haviland-Jones, 2000; Mayne & Ramsey, 2001; Abraham, 1992; Abraham & Gilgen, 1995; Sulis & Combs, 1996; Butz, 1997; Goertzel, 1994; Combs et al., 2003).

Important findings were made in the last decade that were effected from the application of nonlinear dynamic concepts to social sciences (Helbing, 1995). Interpretation of pedestrian dynamics, and social force paradigm, in a framework of molecular dynamics (Helving, 1992), and studies of excitation dynamics in crowds at stadiums in terms of excitable natural systems (Farkas et al., 2002) are among the most energizing results in the field. Attractive parallels between reaction–diffusion chemical systems, morphogenesis, and pattern formation in social insects were built (Bonabeau et al., 1999). Dynamics of opinion formation (Kacperski & Holyst, 1999; Plewczyński, 1998) and attitude change (Nowak, 1990) is yet another field in which techniques of nonlinear physics are flourishing (for example, see Plewczyński, 1998), where social changes in a collective, with a lattice topology, are described in terms of nonlinear Schrödinger equations by analogy with superfluid and weakly interacting Bose gas in an external potential. The nonlinear physics approaches are reinforced by automaton-based computational techniques (Ilachinski, 2001). Good examples are automaton models of artificial societies and the emergence of effective structure in collectives of simple agents and populations of interacting finite automata (Epstein & Axtell, 1996; Ilachinski, 2004). The analogies between the dynamics in natural complex systems and mental processes seem to be promising (Bedau, 1997).

Representation of mind as an agent collective has a long history. We can find artificial intelligence and biological and philosophical backgrounds of the multiagent mind in Minsky's ideas of mind as a society of many agents (Minsky, 1988); Dawkins' memes, ideas, and beliefs, which diffuse in the noosphere and represent some kind of evolving space–time configuration of global mental states (Dawkins, 1976); and Dennet's theory of consciousness as the space–time configuration of a many-agent collective (Dennet, 1991). More computation-oriented results include Wiedermann's cogitoid (Wiedermann, 1998), where knowledge is represented by a lattice of concepts; Hameroff-Penrose's OR theory of consciousness (Hameroff & Penrose, 1996; Hameroff, 1998), where a flow of consciousness is seen as a stream of self-collapsing quantum coherent superpositions; and Alexander's artificial neural consciousness (Alexander, 1997), where consciousness is represented by firing mosaics of automaton-like neurons.

Chemical Paradigm of Artificial Minds

A paradigm of a computational chemistry, where molecules represent computational processes, is first brought into effective action in a context of self-maintenance of system and autopoiesis (Varela et al., 1974, McMullin & Varela, 1997). A molecule in computational chemistry is a symbolic representation of an operator, the molecule's behaviour is the operator's action, and a chemical reaction is an evaluation of the functional application (Fontana & Buss, 1996). A chemical abstract machine, a stirred solution defined by moleculer algebra and rules of molecule interaction, is formalized in the literature (Berry & Boudol, 1992) to realize a paradigm of interaction between molecules in algebraic process calculi. Recently, the artificial chemistry paradigm emerged and formed a new field concentrating on such issues as self-organization, complexity, nonstandard computation (for an overview, see Dittrich et al., 2001), and the application of the chemical paradigm to simulate evolution of knowledge and emotions (Adamatzky, 2001-2003).

Two more topics of artificial life — artificial society and artificial war — are close to the chemical paradigm of artificial minds. Artificial societies deal with the emergence of societal structure in collectives of locally interacting agents (Epstein & Axtell, 1996). The research focuses on automata models of growing societies (Gilbert & Conte, 1995) and populations of interacting finite automata (Axelrod, 1997). A theory of artificial war developed by Andrew Ilachinski (2004) deals with reaction-like interactions between diffusing and colliding living forces.

We simulate stirred mind-reactors using systems of ordinary differential equations and thin-layer (nonstirred) reactors using cellular automata. A cellular automaton is a uniform array of finite automata that update their states in parallel. Each automaton calculates its next state depending on the states of its closest neighbors. We study one-dimensional cellular automata, where each cell updates its state depending on the states of its left and its right neighbors. For example, let a set of reactions be $X + Y \rightarrow 2Z$ and $Z \rightarrow X$. An emotional or doxastic carrier in state X *collides* with a carrier in state Y; they both change their states to affective or doxastic state Z as a result of the collision. In a cellular-automaton model of this medium, every cell has two neighbors, left and right, and takes three states *X*, *Y*, and *Z*. A cell updates its state depending on the states of its two neighbors: a cell being in state *Z* takes state *X*; a cell changes its current state *X*(*Y*) to state *Z* if there is at least one neighbor in state *Y*(*X*). Some cellular-automaton rules are probabilistic, because, due to the reaction equations simulated, a cell takes different states, corresponding to the same configuration of the cell's neighborhood, with certain probabilities.

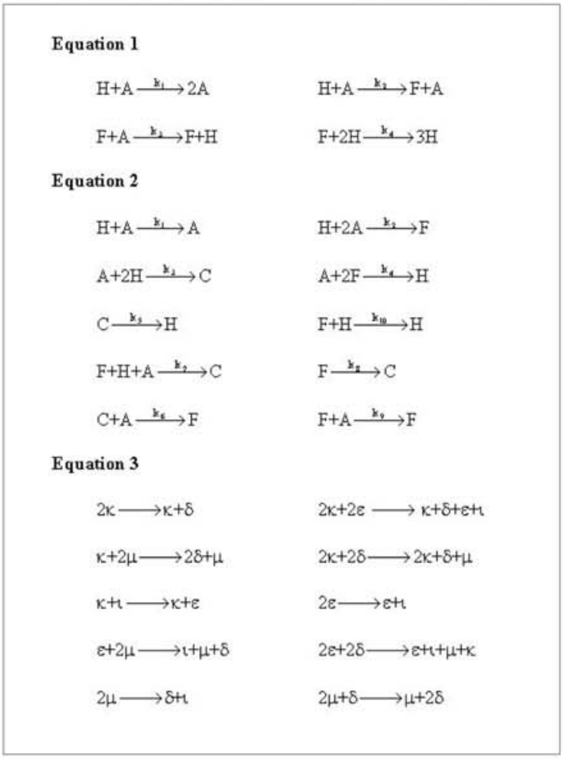
Affective Medium

Emotions interact via their carriers, or agents. Emotions are contagious in a sense that emotional perturbations spread in groups and collectives (Hatfield et al., 1994): "The precipitating stimuli arise from one individual, act upon ... one or more other individuals, and yield corresponding or complementary emotions ... in these individuals." In real life, the contagion may be based on imitation and facial mimicry or a physiological synchronization (Levenson & Ruef, 1997).

A few publications explain not simply a contagion but an interaction between emotions (Hatfield et al., 1994; Levy & Nail, 1993; Marsden, 1998). We know that a person in certain emotional states would probably change his emotions when he encounters other emotional states, for example (Hatfield et al., 1994): "It is happy people who are most receptive to others and most likely to catch their moods; the unhappy seem relatively oblivious to other's feelings and to contagion." This means a table of binary interaction of emotions may have a nontrivial form. Particulars of emotional interaction are usually dependent on the strength of emotions, the context of emotional interaction, and the vectors of interpersonal relationships. We comprise an affective mixture of four types of reactant emotions: happiness (*H*), anger (*A*), fear (*F*), and confusion (*C*). These four are often considered to be basic emotions (Oatley, 1992), because they give us a good sampling of quadrants of the circumplex model of emotions (Russell, 1980) and, thus, are seen as sufficient representatives of a universe of affective states.

Happiness, anger, and fear are basic emotions in a historical context of social studies (McPhail, 1991). According to Hatfield (1994), "Angry faces sometimes stimulate fear as well as anger; another's fear may put us at ease." This determines the first three entries of the reaction scheme in Equation 1 (Figure 1). The fourth reaction shows that fear recovers back to happiness in a *happy* environment. Either anger or fear is the first instant reaction to someone else's anger. Recognition of happiness and fear is not instantaneous and takes some time. Recovery from fear to happiness is a lengthy process. These particulars of nonsymbolic, and often noncommunicative, affective interactions determine exact values of the reaction rates: $k_1 = 0.1$ (aggressiveness, fearlessness); $k_2 = 0.1$ (fearfulness); and $k_3 = 0.09$ and $k_4 = 0.005$ (recoverability).

Figure 1. Examples of quasi-chemical reactions in affective (Equations 1 and 2) and doxastic (Equation 3) media



Equation 1

$H+A \xrightarrow{k_1} 2A$ $H+A \xrightarrow{k_2} F+A$

$F+A \xrightarrow{k_3} F+H$ $F+2H \xrightarrow{k_4} 3H$

Equation 2

$H+A \xrightarrow{k_1} A$ $H+2A \xrightarrow{k_2} F$

$A+2H \xrightarrow{k_3} C$ $A+2F \xrightarrow{k_4} H$

$C \xrightarrow{k_5} H$ $F+H \xrightarrow{k_{10}} H$

$F+H+A \xrightarrow{k_7} C$ $F \xrightarrow{k_8} C$

$C+A \xrightarrow{k_6} F$ $F+A \xrightarrow{k_9} F$

Equation 3

$2\kappa \longrightarrow \kappa+\delta$ $2\kappa+2\varepsilon \longrightarrow \kappa+\delta+\varepsilon+\iota$

$\kappa+2\mu \longrightarrow 2\delta+\mu$ $2\kappa+2\delta \longrightarrow 2\kappa+\delta+\mu$

$\kappa+\iota \longrightarrow \kappa+\varepsilon$ $2\varepsilon \longrightarrow \varepsilon+\iota$

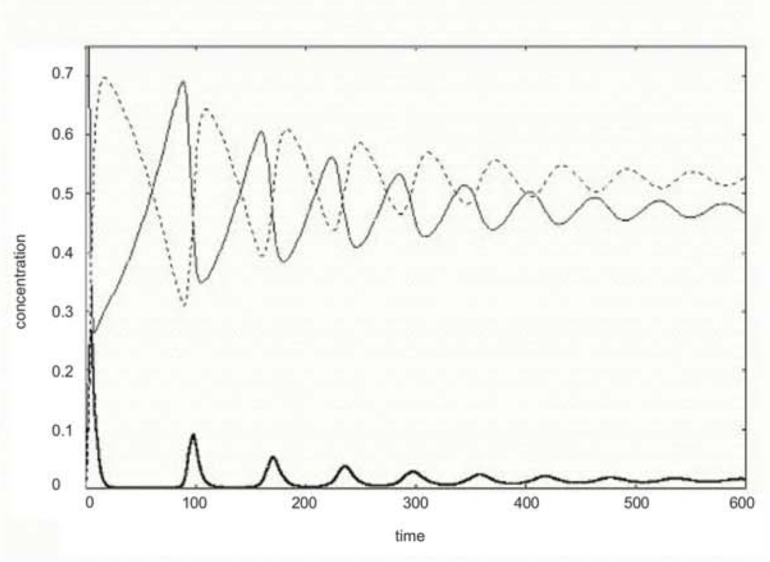
$\varepsilon+2\mu \longrightarrow \iota+\mu+\delta$ $2\varepsilon+2\delta \longrightarrow \varepsilon+\iota+\mu+\kappa$

$2\mu \longrightarrow \delta+\iota$ $2\mu+\delta \longrightarrow \mu+2\delta$

If we add a small quantity of anger to a solution of happiness, governed by Equation 1 (Figure 1), concentrations of all three reactants exhibit damped oscillations, as shown in Figure 2. These findings correspond to oscillations of extreme emotions in experiments with evolving emotional intelligence (Seif El-Nasr et al., 1999). The oscillations per se may demonstrate transient states of mind at the edge of pathology (Plutchik, 2000): "…emotions become symptoms when there is too much of an emotion, when there is too little of an emotion…."

To simulate the affective reactor in cellular automaton, we can simplify the scheme (Figure 1, Equation 1) and substitute the two last entries by the reactions $A \rightarrow H$ and $F \rightarrow H$. Thus, a happy cell takes either anger or fear (with probability 0.5) if one of its neighbors is angry, while angry and fearful cells recover back to the happy state. When a small amount of anger is applied to a solution of happiness, waves of anger and fear are formed: *…H…HAHFH…H…* (wave travels leftward) and *…H…HFHAH…H…* (wave travels rightward). Each

Figure 2. Dynamic of happiness (thin solid line), anger (thick solid line) and fear (dotted line) in a well-stirred reactor with reaction set Equation 1 (Figure 1), initial concentration of happiness is 0.9, anger 0.1, and fear 0



travelling wave has head of anger and tail of fear, so it looks like a typical running impulse. Due to the stochasticity of the cell state transition, the following cell state transitions may occur with the same probability: [*AHA*], [*AH#*], [*#HA*] → {[*A*], [*F*]}, where symbol "#" means *any state*. Therefore, waves may abruptly annihilate (Figure 3). However, the waves rarely cancel each other as a result of collision, rather they form bound states travelling in the medium.

Let us add confusion to the set of reactions and obtain a new reaction scheme, Equation 2 (Figure 1). First and second reactions show how a *flight or fight* choice is made; third and fourth reactions show development of anger. These four reactions suit the framework of evaluation, activation, and potency attributes (Morgan & Heise, 1988), where, for example, local potency of happiness is inversely proportional to concentration of anger in the local vicinity of happiness. Fifth and sixth reactions show that confusion and fear recover to happiness (which is a ground state); seventh and eighth reactions represent recovery of fear to confusion.

What happens if we drop a small quantity of anger in a well-stirred reactor filled with a pure happiness and governed by Equation 2 (Figure 1)? The reactor exhibits a short-time outburst of fear (Figure 4) followed by an outburst of

Figure 3. Space-time dynamics of one-dimensional cellular-automaton model of the reacting happiness, anger and fear; reaction scheme Equation 1 (Figure 1) (The reactor starts its evolution in a uniform state of happiness with a few drops of anger, arbitrarily applied to the medium. Happiness is a solid disc, anger is a circle, and fear is an asterisk. Time arrows downward.)

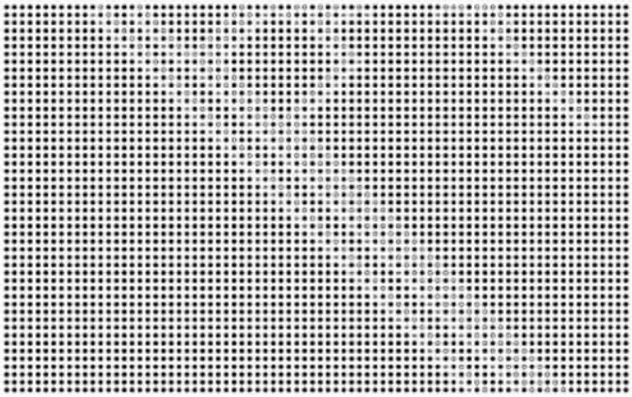


Figure 4. Dynamic of happiness (thick solid line), anger (thin solid line), confusion (dot line) and fear (dot-dash line) in a well-stirred reactor with reaction set Equation 2 (Figure 1), initial concentrations happiness is 0.9, anger 0.1, fear 0, confusion 0; reaction rates are k_1=0.1, k_2=0.1, k_3=0.01, k_4=0.001, k_5=0.001, k_6=0.01, k_7=0.05, k_8=0.001, k_9=0.05, k_{10}=0.01

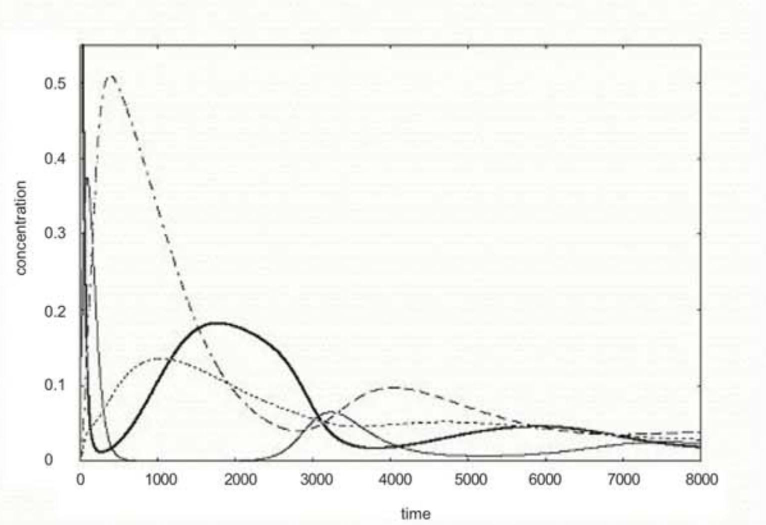


Figure 5. Examples of patterns emerging in cellular-automaton models of affective liquids

....HFACH... ...HCAFH...	....HHHAAFHH... ...HACCCHHH... ...HAAFHHHH... ....
(a)	*(b)*

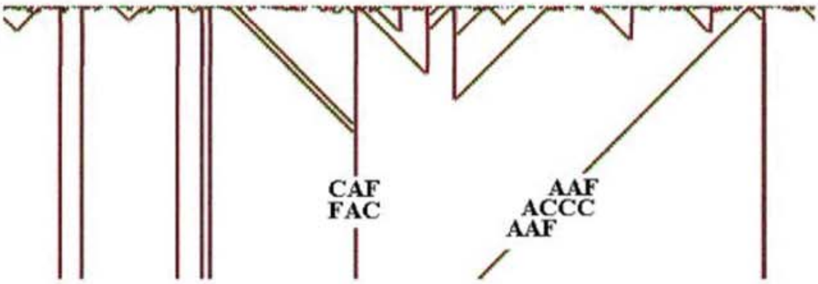
confusion. Happiness decreases in concentration, and recovers later: "*Emotion sometimes seems to come out of nowhere, to hit you with full force, then to be over as quickly as it came … It feels like a freight train of emotion hitting you (or perhaps sweeping you off your feet …*" (Planalp, 1999). These changes in affective concentrations may well be interpreted in a framework of the states of mind stating that ratio of *positive* affects equals 0.63 in healthy people, which roughly corresponds to interval [1000-2000] in Figure 4, and tends to be as low as 0.37 (Garamoni et al., 1991) in patients with pathological depression, as shown after time step 2000 in Figure 4.

Any initial configuration which includes happiness, in a cellular-automaton model, will evolve to a configuration where most cells are in state of happiness and a few sites are occupied by domains of stationary localizations such as shown in Figure 5a, where a domain of anger is supported by neighboring domains alternating between fear and confusion. These domains may be cancelled by travelling localizations (Figure 5b), which themselves switch between anger and confusion and fear while on the move. When an affective medium is locally perturbed by anger, travelling localizations and immobile domains are formed during short transient periods of quasi-chaotic dynamics (Figure 6). Some domains and mobile localizations annihilate in collisions with other travelling patterns. Few domains survive and remain in the system's evolution indefinitely.

The findings show that artifacts of spatiotemporal dynamics, which may be unseen in global-concentration-of-emotions representations, nevertheless possess a huge potential for instantiation of structured intrinsic activity of affective minds. This conforms to the current vision of psychologists (Izard et al., 2000):

"…emotions self-organize as a coherent set or pattern of interacting emotions ... behavioural effects of each emotion, however, may be moderated by the motivational effects of other discrete emotions in the pattern."

Figure 6. Space-time configuration of one-dimensional cellular-automaton model of the reactor Equation 2 (Figure 1), the evolution is started at random configuration of happiness, anger, confusion and sadness (Happy sites are white, others are grey; exact configurations of exemplar stationary and mobile localizations are indicated. Time arrows downward.)



Doxastic Medium

Let us take a belief as a primarily cognitive state, an atom of cognition. The following *second-order* states can be derived from belief:

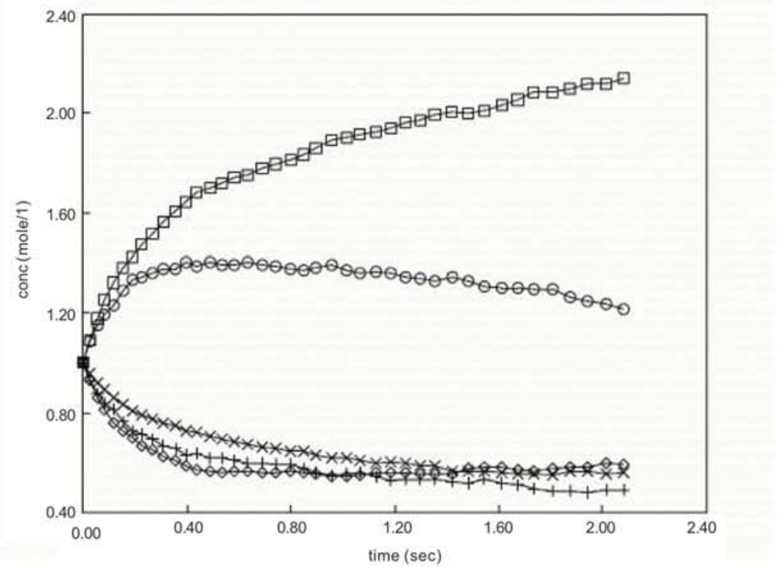
- Knowledge (κ) is the justified belief
- Doubt (δ) is the state of neither belief nor disbelief in a proposition, while the proposition is justified
- Misbelief (μ) is a wrong belief or false opinion
- Delusion (ε) is an erroneous belief or a fixed belief that cannot be justified
- Ignorance (ι) is doubt in a situation when the proposition is not justified

In previous papers (Adamatzky, 2001a, b), we derived a table of binary compositions between the doxastic states (Figure 7). The set of reactions derived from the table is shown in Equation 3 (Figure 1). We use models introduced in this section to derive several basic propositions about the behaviour of doxastic mixtures (Adamatzky, 2001a). Thus, for example, a state of global misbelief is an unreachable state of doxastic mixture. States of global ignorance or global doubt are fixed attracting points of the mixture development. Pure mixtures of delusion, doubt, or ignorance are inert.

Figure 7. Table of binary composition of doxastic states

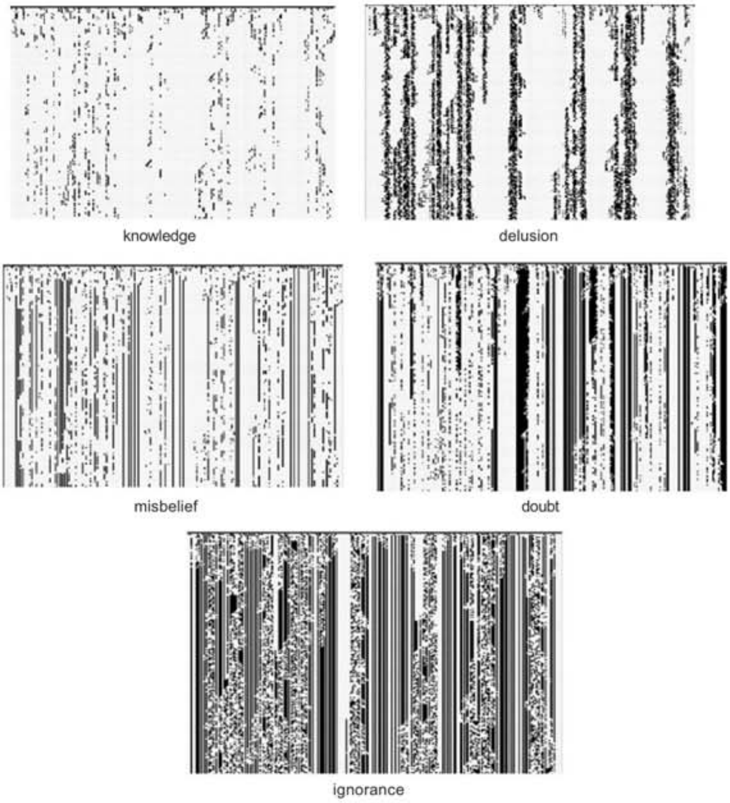
	κ	ε	μ	δ	ι
κ	$\{\kappa, \varepsilon\}$	$\{\kappa, \delta\}$	$\{\delta\}$	$\{\kappa, \delta\}$	$\{\kappa\}$
ε	$\{\varepsilon, \iota\}$	$\{\varepsilon, \iota\}$	$\{\iota\}$	$\{\varepsilon, \iota\}$	$\{\varepsilon\}$
μ	$\{\mu, \delta\}$	$\{\mu, \delta\}$	$\{\delta, \iota\}$	$\{\mu, \delta\}$	$\{\mu\}$
δ	$\{\mu, \kappa\}$	$\{\mu, \kappa\}$	$\{\delta\}$	$\{\delta\}$	$\{\delta\}$
ι	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{\iota\}$	$\{\iota\}$	$\{\iota\}$

Figure 8. Global dynamic of doxastic solution (Equation 3, Figure 1) of doubt (δ), delusion (ε), ignorance (ι), knowledge (κ), and misbelief (μ) in a stirred reactor



Prepare a mixture of all five doxastic reagents, each in the same concentration, and pour the mixture in a continuously stirred reaction vessel. The derived concentrations of doubt and ignorance increase, and concentrations of knowledge, misbelief, and delusion decrease (Figure 8). This happens because doubt is produced when knowledge and misbelief dissociate, and in the reactions between knowledge and misbelief, knowledge and delusion, and misbelief and

Figure 9. Example of space-time evolution of one-dimensional doxastic mixture, initially with randomly distributed reagents (Time arrows down. The configuration is split into five sub-configurations to show dynamics of each of the reagents. For example, in the configuration of knowledge only medium, sites with knowledge are shown in black all others are white.)



delusion. Doubt is consumed when it reacts with delusion. As stated in Frijda and Mesquita (2000), "A belief may persist in the face of evidence that contradicts it. The belief renders that evidence powerless. It is ignored or dismissed; arguing appears useless." The situation with ignorance is similar to doubt, but here doubt contributes to the production of ignorance. This does not contradict common sense, because doubt and ignorance are usual attributes of a proper intellectual.

Does the behaviour of a doxastic mixture change when the mixture is spread in a thin, one-dimensional layer? When initially all reactants are distributed in the layer at random, relatively stable domains of misbelief, doubt, and ignorance emerge (Figure 9). Between these domains, you can find breathing patterns of

Figure 10. Example of space-time evolution of one-dimensional solution of doubt (shown by dots) with a drop of delusion (ε)

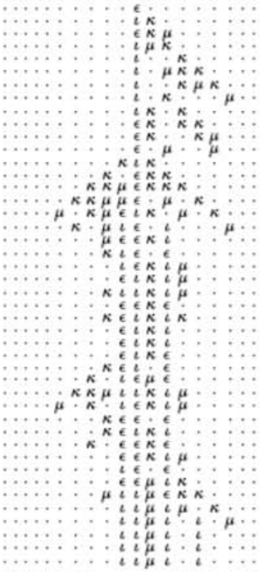
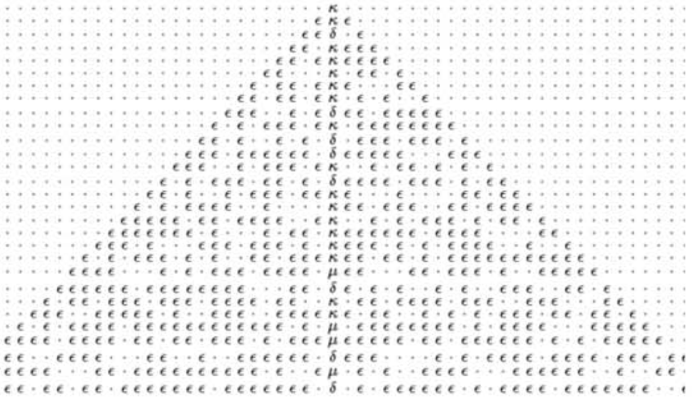


Figure 11. Example of space-time evolution of one-dimensional solution of ignorance (shown by dots) with a drop of knowledge (κ)



knowledge, delusion, and ignorance. The existence of such stable localizations is in line with results of computer experiments with populations of mobile automata competing for resources (Doran, 1998), which demonstrate that a collective misbelief, when coexisting in space and time with collective belief, may increase survivability of the entire population.

When we add a drop of delusion to a solution of doubt, a domain of knowledge, misbelief, and delusion is formed. The domain emits a halo of knowledge and misbelief, these states subsequently die out, and eventually we can see localized domains of ignorance and doubt (Figure 10).

In contrast to the previous example, adding knowledge to a thin-layer mixture of ignorance leads to catastrophic changes in doxastic mixture. A localization of knowledge intermitting with doubt is formed, and it emits quasi-waves of delusion that spread across the mixture (Figure 11). This may well characterize a transitional period from a normal state of mind to a global delirium, when a bizarre delusion is initiated by a clouded state of mind (Charlton, 2003):

> *The release from a normal coherent progression to a quasi-pathological and unpredictable association of ideas varies in severity on a continuum from occasional lapses of concentration to gross incoherence. The process can be observed by other people when a delirious patient exhibits a fluctuating state of consciousness, lucid and rational intervals interspersing drowsy or agitated periods of illogical thought.*

Conclusion

In this chapter, we exemplified an artificial chemistry based theory of mind and discussed characteristic findings we obtained in computational experiments with doxastic and affective solutions. We showed that certain phenomena of cognitive and emotional developments can be demonstrated in constructively simple and behaviorally rich models of quasi-chemical reactors.

Fields of potential application that will benefit from the development of the parachemistry of mind include artificial consciousness of agent collectives; hard- and soft-consciousness of computers and robots, for example, the implementation of chemical models of consciousness in programs and silicon processors; and, reaction-diffusion consciousness of artificial organic creatures, for example, liquid chemical brains, diffusive intelligence, and cognitive amoeboids. Our pilot experiments with the control of robot navigation (Adamatzky et al., 2004) and artificial hand (Yokoi et al., 2004) using experimental reaction-diffusion chemical processors demonstrated the feasibility of the approach. The next step would be to derive more accurate correspondence between emotional and cognitive processes and real-life chemical reactions. The interpretation of mental processes in terms of artificial chemical systems will hopefully allow us to simulate a wide range of abnormal states of mind, from narcissism (Dimaggio et al., 2002) to delusional disorders (Charlton, 2003).

Using cellular-automaton techniques, we demonstrated the importance of structural dynamics in doxastic and affective solutions. This is ideologically similar to studies in individual attitude change (PlewczyD´ski, 1998), where classical mathematical models lead to entire uniformity of opinions in collectives, while local interaction-based models allows for the representation of space-time dynamics of individual opinions and opinion pattern formation, for example, the stable clusters of individuals sharing minority opinions.

What are possible future developments in the field? We deliberately took affective and doxastic species out of their natural habitats of intentions, actions, interpersonal relations, achievements, and social norms. In further studies, the computational models ought to be enriched with real-life traits. For example, emotions do not simply determine a social structure but synchronize social dynamics and govern formation of coherent actions (Heise & O'Brian, 1993; Planalp, 1999). Therefore, we should aim to couple affective solutions with effective solutions to develop novel reactions schemes.

We also did not discuss interactions between emotions and beliefs, because the dynamics of affective and doxastic liquids were set apart from one another. Mixing these two solutions together would be a great future challenge (Frijda et al., 2000): "… thinking, no matter how well articulated, is not sufficient for action. … Emotions are prime candidates for turning a thinking being into an actor." Emotions are indispensable not only in cognition but also in the development of social structure (Barbalet, 2000). This may be of particular importance to studies in collective mind, as for example, in crowds, where people increase their emotional reactions, decrease intellectual reactions, and experience growth of rationalization and irresponsibility (Moscovici, 1986).

References

Abraham, F. D. (1992). Chaos, bifurcations, and self-organization: Dynamical extension of neurological positivism and ecological psychology. *Psychoscience, 1*, 85–118.

Abraham, F. D., & Gilgen, A. R. (1995). *Chaos theory in psychology*. Westport, CT: Greenwood Press.

Adamatzky, A. (2001). Chemistry of belief: Experiments with doxastic solutions. *Kybernetes, 30*, 1199–1208.

Adamatzky, A. (2001a). Space–time dynamic of normalized doxatons: Automata models of pathological collective mentality. *Chaos, Solitons and Fractals, 12*, 1229–1256.

Adamatzky, A. (2001b). Pathology of collective doxa. Automata models. *Applied Mathematics and Computation, 122*, 195–228.

Adamatzky, A. (2001c). *Computing in nonlinear media and automata collectives*. Bristol; Philadelphia: Institute of Physics Publishing.

Adamatzky, A. (2002). On dynamics of affective liquids. *Dynamical Psychology*.

Adamatzky, A. (2003). On patterns in affective media. *International Journal of Modern Physics C, 14*, 673–687.

Adamatzky, A. (2003a). Affectons: Automata models of emotional interactions. *Applied Mathematics and Computation, 146*(2), 579–594.

Adamatzky, A. (2004). *Nonlinear dynamics of crowded minds*, forthcoming.

Adamatzky, A., de Lacy Costello, B., Melhuish, C., & Ratcliffe, N. (2004). Experimental implementation of mobile robot taxis with onboard Belousov-Zhabotinsky chemical medium. *Materials Science and Engineering C*, submitted.

Alexander, I. (1997). *Impossible minds: My neurons, my consciousness*. Singapore: World Scientific Publishing.

Axelrod, R. (1997). *The complexity of cooperation: Agent-based models of competition and collaboration.* Princeton, NJ: Princeton University Press.

Barbalet, J. M. (2000). *Emotion, social theory, and social structure*. London; New York: Cambridge University Press.

Bedau, M. A. (1997). Emergent models of supple dynamics in life and mind. *Brain and Cognition, 34*, 5–27.

Bellman, R. (1967). Mathematical models of the mind. *Mathematical Biosciences, 1*, 287–304.

Berry, G., & Boudol, G. (1992). The chemical abstract machine. *Theoretical Computer Science, 96*, 217–248.

Bonabeau, E., Dorigo, M., & Theraulaz, G. (1999). *Swarm intelligence: From natural to artificial systems*. Oxford: Oxford University Press.

Butz, M. R. (1997). *Chaos and complexity: Implications for psychological theory and practice*. London: Taylor & Francis.

Charlton, B. G. (2003). Theory of Mind delusions and bizarre delusions in an evolutionary perspective: Psychiatry and the social brain. In M. Brune, H. Ribbert, & W. Schiefenhovel (Eds.), *The social brain — Evolution and pathology*. New York: John Wiley & Sons.

Combs, A., Germine, M., & Goertzel, B. (Eds.). (2003). *Mind in time: The dynamics of thought, reality, and consciousness*. Australia: Hampton Press.

Dawkins, R. (1976). *The selfish gene*. Oxford: Oxford University Press.

Dennet, D. (1991) *Consciousness explained*. London: Penguin.

Dimaggio, G., Semerari, A., Falcone, M., Nicolò, G., Carcione, A., & Procacci, M. (2002). Metacognition, states of mind, cognitive biases, and interpersonal cycles: Proposal for an integrated narcissism model. *Journal of Psychotherapy Integration, 12*, 421–451.

Dittrich, P., Ziegler, J., & Banzhaf, W. (2001). Artificial chemistries: A review. *Artificial Life, 7*, 225–275.

Doran, J. (1998). Simulation collective misbelieve. *Journal of Artificial Societies and Social Simulation,* 1. Retrieved from the World Wide Web: *http://www.soc.surrey.ac.uk/JASSS/1/1/3.html*

Epstein, J. M., & Axtell, R. L. (1996). *Growing artificial societies*. Cambridge, MA: MIT Press.

Farkas, I., Helbing, D., & Vicsek, T. (2002). Mexican waves in an excitable medium. *Nature, 419*, 131–132.

Fontata, W., & Buss, L. W. (1996). The barrier of objects: From dynamical systems to bounded organizations. *SFI Working Paper*. 96-05-035. Santa Fe, NM: Santa Fe Institute.

Frijda, N. H., & Mesquita, B. (2000). Beliefs through emotions. In N. H. Frijda, A. S. R. Manstead, & S. Bem (Eds.), *Emotions and beliefs. How feelings influence thoughts* (pp. 45–77). London; New York: Cambridge University Press.

Frijda, N. H., Manstead, A. S. R., & Bem, S. (2000). The influence of emotions on beliefs. In N. H. Frijda, A. S. R. Manstead, & S. Bem (Eds.), *Emotions and beliefs. How feelings influence thoughts* (pp. 1–10). London; New York: Cambridge University Press.

Garamoni, G. L., Reynolds, C. F., Thase, M. E., Frank, E., Berman, S. R., & Fasiczka, A. L. (1991). The balance of positive and negative affects in major depression: A further test of the States of Mind model. *Psychiatry Research, 39*, 99–108.

Gilbert, N., & Conte, R. (1995). *Artificial societies: The computer simulation of social life*. London: UCL Press.

Goertzel, B. (1994). *Chaotic logic: Language, thought, and reality from the perspective of complex systems science*. New York: Plenum Press.

Goleman, D. (1996). *Emotional intelligence*. London: Bloomsbury Publishing.

Hameroff, S. (1998). Quantum computation in brain microtubules? The Penrose–Hameroff Orch OR model of consciousness. *Philosophical Transactions Royal Society London A, 356*, 1869–1896.

Hameroff, S. (2001). Consciousness, the brain, and spacetime geometry. *Annals of the New York Academy of Science, 929*, 74–104.

Hameroff, S., & Penrose, R. (1996). Conscious events as orchestrated space–time selections. *Journal of Consciousness Studies, 2*, 36–53.

Hatfield, E., Cacioppo, J. T., & Rapson, R. L. (1994). *Emotional contagion*. London; New York: Cambridge University Press.

Heise, D. R., & O'Brien, J. (1993). Emotion expression in groups. In M. Lewis, & J. M. Haviland (Eds.), *Handbook of emotions* (pp. 489–497). New York: Guilford Press.

Helbing, D. (1992). A fluid-dynamic model for the behaviour of pedestrians. *Complex Systems, 6*, 391–415.

Helbing, D. (1995). *Quantitative sociodynamics*. Dordrecht: Kluwer Academic.

Ilachinski, A. (2001). *Cellular automata: A discrete universe*. Singapore: World Scientific Publishing.

Ilachinski, A. (2004). *Artificial war: Multigent-based simulation of combat*. Singapore: World Scientific Publishing.

Izard, C. E., Ackerman, B. P., Schoff, K. M., & Fine, S. E. (2000). Self-organization of discrete emotions, emotion patterns, and emotion-cognition relations. In M. D. Lewis, & I. Granic (Eds.), *Emotions, development and self-organization* (pp. 15–36). London; New York: Cambridge University Press.

Kacperski, K., & Holyst, J. A. (1999). Opinion formation model with strong leader and external impact: A mean field approach. *Physica A, 269*, 511–526.

Lettvin, J. Y., & Pitts, W. (1943). A mathematical theory of the affective psychoses. *Bulletin of Mathematical Biophysics, 5*, 139–148.

Levenson, R. W., & Ruef, A. M. (1997). Physiological aspects of emotional knowledge and rapport. In W. J. Ickes (Ed.). *Emphatic accuracy* (pp. 44–72). New York: Guilford Press.

Levy, D., & Nail, P. R. (1993). Contagion: A theoretical and empirical review and reconceptualization. *Social and General Psychology Monographs, 119*, 183–235.

Lewis, M., & Haviland-Jones, J. M. (Eds.). (2000). *Handbook of emotions*. New York: Guilford Press.

Lewis, M. D., & Granic, I. (Eds.). (2000). *Emotion, development, and self-organization: Dynamic systems approaches to emotional development*. London; New York: Cambridge University Press.

Marsden, P. (1998). Memetics and social contagion: Two sides of the same coin. *Journal of Memetics, 2*.

Mayne, T. J., & Ramsey, J. (2001). The structure of emotions: A nonlinear dynamic systems approach. In T. J. Mayne, & G. Bonanno (Eds.), *Emotions: Current issues and future directions. Emotions and social behaviour* (pp. 1–37). New York: Guilford Press.

McMullin, B., & Varela, F. J. (1997). Rediscovering computational autopoiesis. *SFI Working Paper*, Working Papers 97-02-012. Santa Fe, NM: Santa Fe Institute.

McPhail, C. (1991). *The myth of the madding crowd*. Berlin: Aldine de Gruyter.

Mayne, T. J., & Ramsey, J. (2001). The structure of emotions: A nonlinear dynamic systems approach. In T. J. Mayne, & G. Bonanno (Eds.), *Emotions: Currrent issues and future directions. Emotions and social behavior* (pp. 1–37). New York: Guilford Press.

Minsky, M. L. (1988). *The society of mind*. New York: Simon & Schuster.

Morgan, R. L., & Heise, D. (1988). Structure of emotions. *Social Psychology Quarterly, 51*, 19–31.

Moscovici, S. (1986). The discovery of masses. In C. F. Graumann, & S. Moscovici (Eds.), *Changing conceptions of crowd mind and behaviour* (pp. 5–25). Heidelberg: Springer-Verlag.

Nowak, A., Szamrej, J., & Latane, B. (1990). From private attitude to public opinion: A dynamic theory of social impact. *Psychological Review, 97*, 362–376.

Oatley, K. (1992). *Best laid schemes: The psychology of the emotions*. London; New York: Cambridge University Press.

Planalp, S. (1999). *Communicating emotion*. London; New York: Cambridge University Press.

Plewczyński, D. (1998). Landau theory of social clustering. *Physics A, 261*, 608–617.

Plutchik, R. (2000). *Emotions in the practice of psychotherapy*. Washington, D.C.: American Psychological Association.

Russell, J. A. (1980). A circumplex model of affect. Jo*urnal of Personality and Social Psychology, 36*, 1161–1178.

Seif El-Nasr, M., Ioerger, T. R., & Yen, J. (1999). A web of emotions. In *Proceedings of Workshop on Emotion-Based Agent Architectures* (part of Autonomous Agents '99).

Stauffer, D. (2003). Sociophysics simulations. *IEEE Computing in Science & Engineering, 5*(3), 71–75.

Sulis, W., & Combs, A. (Eds.). (1996). *Nonlinear dynamics in human behaviour*. Singapore: World Scientific Publishers.

Varela, F. J., Maturana, H. R., & Uribe, R. (1974). Autopoiesis: The organization of living systems, its characterization and a model. *BioSystems, 5*, 187–196.

Wiedermann, J. (1998). Towards algorithmic explanation of mind evolution and functioning. In L. Brim et al. (Eds.), *Proceedings of MFCS'98. Lecture Notes in Computer Science, 1450*, 152–166.

Yokoi, H., Adamatzky, A., De Lacy Costello, B., & Melhuish, C. (2004). Excitable chemical medium controlled for a robotic hand: Closed loop experiments. *International Journal of Bifurcation and Chaos*, in press.

Chapter 2

Beyond Needs: Emotions and the Commitments Requirement

Michel Aubé
Université de Sherbrooke, Canada

Abstract

This chapter proposes a model of emotions relying upon an analysis of the requirements that are to be met by individuals of nurturing species, so as to adapt themselves to their social environments. It closely reflects the structure of other motivational systems, which consist of control structures dedicated to the management of resources critical for survival. The particular resources emotional systems seem to handle have to do with social bonding and collaborative behaviors. Herein, they are called second-order resources. They refer to the resources made available by other agents, and they are captured in the model through the concept of commitments. Emotions thus appear as computational control systems that handle the variation of commitments lying at the root of interactive and collaborative behaviors. Some critical consequences of the model for the implementation of emotions in artificial systems are drawn at the end of the chapter.

Introduction

In modelling the mind, Aaron Sloman (1987, 1995; Beaudoin & Sloman, 1993) used to say, "architecture is more important than design." He also contended that a robust computational architecture had to stem from an adaptive mapping from *requirement space* onto *design space*. Indeed, for any computational system to *survive* or succeed in a given environment, there is a set of requirements to be met. To meet these requirements, there is a space of possible designs that could more or less satisfy the constraints they exert on survival or success. Mind modelling thus implies that a reasonable set of requirements be clearly specified for a given *species* (or class of systems) and that a variety of possible designs be explored as reasonable solutions for handling those constraints. Perceptual and memory systems as well as motor routines, habits, and organized behaviors, count among the myriad of designs that emerged through natural selection to meet the requirements of adaptation and that compose the overall architecture of the mind. For some of these processes, researchers in artificial intelligence, robotics, and cognitive science specified a list of requirements that commanded their existence and proposed reasonable designs to handle them (Maes, 1991; Steels & Brooks, 1995).

The following chapter will consider emotional systems along these lines and argue that they probably also developed through evolution as a class of designs slowly carved to handle particular kinds of requirements. It is the author's opinion that in most current computational models of the mind, emotions have unfortunately been merged with a whole bunch of motivational processes that differ in kinds, functions, and levels of complexity. This confusion has obscured their successful understanding and implementation. Yet, such an assertion does not presuppose that emotions emerged at about the same evolutionary period or even in close succession. Just as the different perceptual systems probably emerged from random mutations, provided they offered adaptations significant enough to be selected, so the different emotional systems probably also issued from random mutations that proved successful for concerned individuals, given the requirements they had to face and meet in their own ecosystems.

Even though perceptual systems did not emerge from a well-delineated plan, it is profitable for an engineer to conceive of them as *energy transducers* that could provide various kinds of information for a given artefact. This idea directly follows from an analysis of the requirements involved: (1) autonomous agents need information to act upon their environment and move around it; (2) objects and events, by their presence and occurrence, bear consequences upon their surroundings and thus perturb the energy flow around them (light, heat, sound, pressure); (3) measuring sudden variations of energy around oneself provides a way of capturing relevant information about current events. Estimating different types of energy gradients makes for the diversity of sensory systems and, hence, contributes to the richness of the information gathered. Such a theoretical view certainly helps engineers understand how perception is made possible, and design and implement various perceptual systems in animats.

In a similar way, we are to propose that emotional systems could be envisioned as belonging to a class of designs built around certain shared principles to fulfil certain types of functions. We contend that the requirement analysis goes a long way to help clarify the general characteristics of emotions, and to specify the constraints that their particular structure has been evolutionarily shaped to handle. Consequently, it should prove useful for robotics engineers and cognitive scientists to identify and understand these common principles. Our argument to capture and specify the structure of emotions is threefold:

1. Emotions, just like needs, clearly belong to motivational processes; motivation essentially has to do with resources management.
2. There are strong reasons to believe that emotions mainly have to do with interactions between individuals.
3. The particular resources that emotions (as motivational systems) seem to handle have to do with setting up, monitoring, and managing relations between individuals. To this particular kind of resources, we gave the name *commitment*, herein defined as a mental disposition—though not necessarily conscious—to cooperate with and care about other partners, so as to build durable and fruitful relations with them.

In the following section we will first have to clarify a few points about emotions as *motivational processes*. Second, we will stress the *interactive character* of emotions and show the importance this bears upon their nature and functions. Thereafter, we will present the *concept of commitment*, which plays a central and critical role in the present model. The last part will enumerate some *critical consequences* issued from this point of view, notably about the implementation of emotional processes within intelligent artifacts.

Emotions Belong to the Motivational

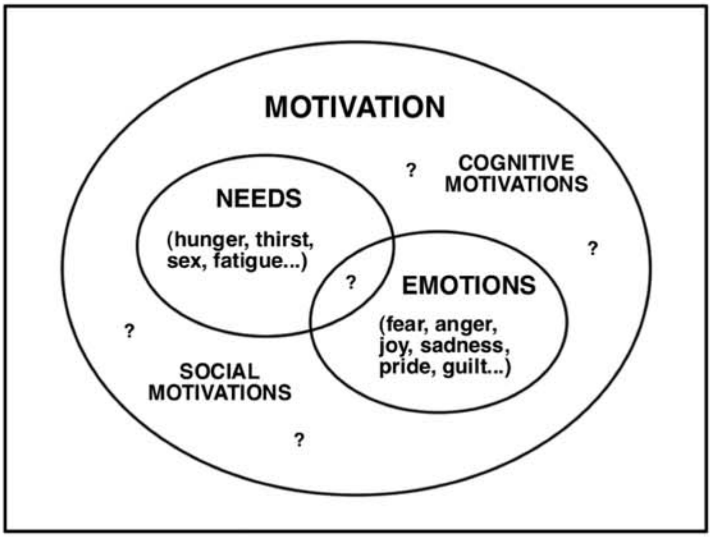
First, we must specify what constitutes emotions. Great progress has been made in the understanding of emotions by adopting a cognitive point of view. Yet, emotions are not the same kinds of entities as are other cognitive processes, such as perception and memory, or planning and problem solving, and they do not lie on the same plane of reference as these other cognitive processes. A simple metaphor might help to illustrate the point. Think about a relatively autonomous robot—somewhat like *Sojourner*—sent on a distant planet to gather a variety of data for humans. By definition, such a mechanism will have to fulfil a certain amount of prescribed tasks according to the planned mission. *But its foremost goals will not be about the objectives of the mission, they will have to be about survival and about remaining a functional device throughout.*

The cognitive analysis is useful, even mandatory, in understanding how perception and memory work, and how they provide an animat with a workable representation of its world. But what is made of this representation in deciding where to go and what to do next, so as to stay *alive*, is another matter. *Stake is the very stuff autonomy is made of in organisms.* Cognition is about the way things are, or rather the way they could be represented so as to enable one to act upon them. But the reasons to act upon them in certain ways are not prescribed by the nature of things and have to come from somewhere else. *There has to be a stake behind autonomous decisions, and motivation is precisely about stake*. What motivates a living organism has less to do with logic than with history. It stems from the evolutionary path to the species to which this individual belongs, it has to be wired-in, and it is basically rooted in survival, through a chain of subgoals of variable lengths (Barkow, Cosmides, & Tooby, 1992; Pinker, 1997).

Most theorists of emotions compare emotions with needs, such as hunger, thirst, or fatigue. Emotions appear as motivators—they seem to be endowed with the strength and capacity of making us do things we would not have thought of doing otherwise. Psychologists typically see the motivation construct as the psychic force that lies behind and gives sense to changes in behavior (Vallerand & Thill, 1993). Motivational systems play against psychic inertia. They have evolved to make organisms change from behavior to behavior when it is presumably more adaptive to do so. Some of these internal forces are more of a physiological kind, such as needs (hunger, thirst, sex, or fatigue), while others are more of a psychological kind, such as emotions (anger, fear, sadness, or joy). These two types of motivational systems are those most thoroughly studied so far, yet they probably do not exhaust all of the internal forces that determine behavior. As suggested in Figure 1, they might not suffice to fill the whole motivational field by themselves.

Many emotion theorists would, for instance, reject surprise and curiosity as emotions proper (Johnson-Laird & Oatley, 1989; Ortony, Clore, & Collins, 1988; Stein, Trabasso, & Liwag, 1993), but such cognitive tendencies act so as to motivate us when we are confronted with novel information. Likewise, persistence in pursuing goals we launched, is certainly motivational, although more cognitive. Researchers in the field of social psychology also identify forces that

Figure 1. Motivational field

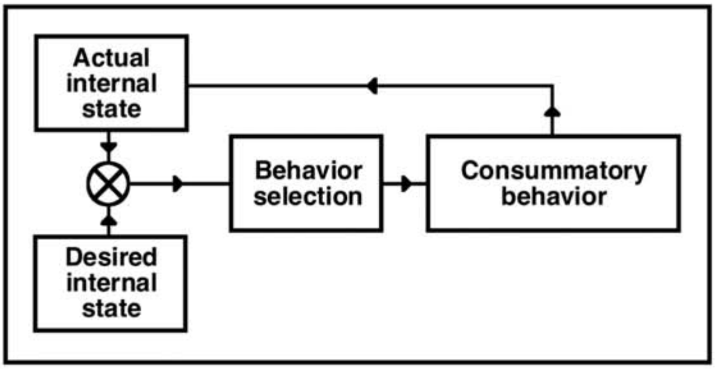

MOTIVATION
NEEDS
(hunger, thirst, sex, fatigue...)
?
COGNITIVE MOTIVATIONS
?
EMOTIONS
(fear, anger, joy, sadness, pride, guilt...)
?
?
SOCIAL MOTIVATIONS
?

clearly determine the behaviors of individuals when they find themselves in small groups (Aronson, 1996; Cialdini, 1984). Yet, it remains possible that such forces actually follow from the dynamics of emotions, especially if they critically fulfil social functions, as we will argue later. In any case, the interrogation marks in Figure 1 make for the possibility that other kinds of motivations are eventually to be uncovered. We also inscribed a mark at the intersection of needs and emotions to reflect the fact that some theorists, such as Minsky (1985), would lump hunger and thirst under the emotion label, while others, such as Toates (1986), would include fear, distress, and aggression in needs. We think that these two sets of behaviors, together with the mechanisms that underlie them, are distinct. The confusion might come from the fact that there are, for instance, different kinds of fears, resting upon different physiological and psychological processes, some simpler and more similar to the structure of needs, and some more complex, unfolding to the full complexity of emotions proper. The important point here is that *need-fear* meets different requirements than does *emotion-fear*, and that they differ in the kind of design that enables them to play their parts.

Now, defining emotions as motivations is not just a lexical matter. It tells something useful about their structures, their functions, their basic dynamics, and, as for the structure of perceptual systems identified above, about the ways they could eventually be implemented in artifacts. For one thing, motivational systems should be seen as *control structures*. Their basic roles are not so much to compute information and provide representations about the world as they are to organize and structure behavior so as to insure survival and adaptation. Of course, this does not preclude that other systems might eventually extract useful representations from their operations. This explains why motivational systems require a high priority of execution, why their algorithmic structures should remain simple and modular, why they frequently operate with emergency, most often on the basis of sparse and incomplete information, and also why it usually gets dramatic whenever they miss their goals. Another essential point about motivational systems has to do with managing resources critical for survival. For instance, a need system such as hunger is built so as to detect deficits in nutriments and to set the whole organism to look for the appropriate resources. Need systems are understood well enough in biology and psychology (Lindsay & Norman, 1972; Toates, 1986). They basically consist of feedback systems that monitor specific resources and trigger certain behaviors when it appears that these resources must be replenished. As a first approximation, we propose that all kinds of motivations rest on a control loop structure, like the one depicted in Figure 2.

An organism requires a certain amount of resources for its survival (water, nutriments, shelter), and these resources are represented by inner variables that compose the internal state of the organism. The flow of variation of these variables is recurrently compared with the desired state, and corrections are imposed whenever a critical deviation is detected. Such regulation typically calls for selecting certain behaviors and executing consummatory activities. As a kind of motivation, emotions appear close enough to needs, and it seemed a good strategy to postulate that they rested on a similar structure. They determine and control a large spectrum of behaviors, they are triggered quickly, and they often go astray. But if emotions operate along the regulatory structure of motivational systems, a fundamental question as to *the kind of resources they manage and regulate* is proposed. The evolutionary tone we adopted so far also suggests that these resources must be precious and critical enough so that, in spite of the costs incurred, they were worth the selection of an additional layer of design.

Figure 2. Basic structure of motivational systems (after Toates, 1986)



The Interactive Character of Emotion

One way of capturing the nature and functions of emotions is to examine the contexts within which they are most likely to arise. On this ground, there are many independent yet converging indications that emotional episodes are largely *interactive* in character. These reasons are regrouped here along three main arguments. The first has to do with expressions, which appear as essential components of emotions. We will propose that expressions are not just side effects that burst out when emotions are too intense, but that they constitute the means by which these motivational systems manage to fulfil their regulatory functions. The second argument has to do with the antecedents of emotions, the recurrent universal situations within which emotions are typically triggered, across human cultures, and even across many animal species. The last argument relies on the fact that emotional behaviors are associated with the operation of the limbic structures of the brain, some of which are considered responsible for the emergence in evolution of maternal behavior, distress calls, and parent-offspring attachment.

All emotions seem to incorporate an expressive aspect, out of which the feeling does not seem complete (Ekman, 1982, 1984; Evans, 2002; Frijda, 1989). It is clearly the case for laughing (Provine, 2000), crying (Lutz, 2001), empathic distress (Levenson & Ruef, 1992; Sagi & Hoffman, 1976), shame and guilt (Baumeister, Stillwell, & Heatherton, 1994; Lewis, 1993; Tangney, 1995), fear (Hoffman, 1974; Rosenblum & Alpert, 1974), and anger (Averill, 1979, 1983; Lemerise & Dodge, 1993). Expressions are very hard to control, and some of the muscles involved in certain expressions—such as the *true smile*—depend on the autonomic nervous system, out of reach of voluntary control. It is as if expressions were intrinsic parts of emotion, and when they are too successfully retained, the feeling almost dies away (Ekman & Davidson, 1994, "Question 7: Can we control our emotions?", pp. 263–281). Their control thus remains one of the most successful paths to the socialization of emotions and to the mastering of display rules, not only because it helps to hide them, but also because it often largely contributes to suppress them (Ekman, 1993; Malatesta & Haviland, 1982). Reciprocally, actors know that putting up the face or posture of a given emotion is a powerful way of eliciting it (Bloch, 1989; Bloch, Orthous, & Santibanez-H, 1987). The effect is strong enough that merely asking a subject to contract the various muscles involved in a given emotional expression has a significant effect in triggering the corresponding emotional experience (Ekman, Levenson, & Friesen, 1983; Larsen, Kasimatis, & Frey, 1992; Strack, Martin, & Stepper, 1988). Finally, individuals in species that have emotions are all subject to strong contagions from the expression of emotions in others (Dimberg, 1982; Hatfield, Cacioppo, & Rapson, 1994; Klinnert, Campos, Sorce, Emde, & Svejda, 1983). This suggests that emotions might have emerged as powerful communicative devices (Oatley & Johnson-Laird, 1996), presumably designed to resolve certain survival problems precisely through such acts of communication.

Researchers from linguistics, philosophy of language, and artificial intelligence suggested that transactions among intelligent agents are set up and managed through the operation of *speech acts*. These powerful language mechanisms play the roles of conveying intentions and setting up commitments between speaking partners involved in coordinated actions (Austin, 1962; Cohen & Levesque, 1990; Cohen, Morgan, & Pollack, 1990; Cohen & Perrault, 1979; Searle, 1969, 1979; Winograd & Flores, 1986). We propose here that an essential part of emotional systems rests precisely upon similar though much simpler *emotional acts* that convey intentions efficiently between partners. For instance, through its anger expression, an animal is signalling to an intruder that the intruder transgressed its territory or came too close to its offspring, and by the same way, its warning conveys some of the intentions of the disturbed partner: "Get off my way, or I may charge you!"

Another intriguing indication of the interactive character of emotion is that most theorists in the field will recognize that emotions are more frequently triggered in situations of encounters, that they are also more intense when interpersonal relationships are involved and are yet more intense when these relations are intimate. Herbert Simon (1967) mentioned this in his seminal paper, "Motivational and Emotional Controls of Cognition," and many psychological experiments and surveys have amply confirmed this. These converging data come from diverse sources: content analyses of individual accounts about emotional episodes (Boucher, 1983); large surveys on emotional experiences gathered from more than a dozen cultures (Scherer, 1988; Scherer, Wallbott, & Summerfield, 1986); ethnological studies (Lutz, 1988); and clinical interviews with children (Stein et al., 1993). The regularity of results for the emotions considered is striking. Thus, *joy* typically stems from relationships with close friends and family members, and it is best expressed in meetings and celebrations with them. *Sadness*, on the other hand, is often caused by the illnesses or deaths of persons close to our hearts, or by problems and difficulties we encounter in our relations with them. *Anger* is everywhere triggered by the transgression of social norms or personal promises, and it is more intense and violent when the transgressors are closer to us. Even in animals, as alluded to above, anger is most often provoked by such transgressions as territory violation, mate steeling, or threat to offspring. Finally, *fear* arises from danger, physical threat and aggression, or risk of social humiliation. As is the case for other emotions, it is experienced almost as intensely when it happens to persons close to us, as when we are concerned, ourselves. There are other causes, which could vary from culture to culture, but those listed above are the most frequently evoked, and the most largely spread out across people and even across other species. As to the effects of cultural variables, they seem to modulate rather than modify the antecedents of emotions. In the case of anger, for example, transgression remains the most common cause, but culture—and perhaps also genes in the case of certain species—determines what is deciphered and considered a transgression.

Finally, some neuroscientists following Paul MacLean (1993; Damasio, 1994, 1999, 2003; LeDoux, 1998; Panksepp, 1991) contend that emotions largely depend upon limbic structures, but also that maternal behavior, distress cries, imprinting, and attachment appeared in evolution conjointly with those neural structures. We mentioned that such a reaction as fear could belong to the need layer or to the emotion layer, depending upon the complexity of its design and the requirements it is apt to meet. The apparition of distress cries and of alarm calls sets the stage for *emotion-fear*. Just consider that yelling and crying in offspring would be maladaptive and counterproductive for individuals of species that could not count on parental attachment and on ferocious protection on their part. It would render them more vulnerable by signalling their presence to predators. Your projected shadow would likely make a frog flee and jump into water, but once caught, it will stay calmly in your hands unless you touch it or make a sudden move. If you try the same with a baby bird or a baby squirrel, their cries and yells will probably have the effect of launching a full parental attack against you. Many psychologists and ethologists following Bowlby (1969, 1973, 1980; Ainsworth, 1989) and Eibl-Eibesfeldt (1975) further suggest, from the study of rituals in adult encounters, that *parent-offspring attachment might be the paradigm scenario of all successive collaborative behaviors.*

Put together, these arguments point to the idea that the resources emotions are designed to handle have to do with bonding, belonging, and collaborative behaviors. Clearly, many emotions could arise when we are alone. But bonding and collaboration continue to operate in our minds and behaviors when friends or relatives are gone. These considerations lead us to envision the "*need to belong*" (Baumeister & Leary, 1995) as a necessary element for the understanding of emotions and probably even as a core component of what they are, and what they are for (Aubé & Senteni, 1996b). The argument about expression also strongly suggests that emotions manage to play their part, and restore the resources they are responsible for, mainly through acts of communication.

Commitments as Second-Order Resources

Clearly, access to resources critical for survival—those that needs are designed to regulate—would be much facilitated if one could count on partners. A pack of reindeer, for example, could more easily protect its young against predators by presenting a common front. Reciprocally, it would be more complicated for a wolf to catch a deer if it had to chase alone. Profiting from other agents thus offers a very precious resource, but at the same time, it raises a complex problem. Cooperation in a selfish Darwinian world is a risky business, wherein profiteers could take all the benefit of collaboration without returning the favor (Axelrod, 1984). So, there is one subtlety here: the other agents could not constitute by themselves the resources which cooperation makes available. What appears critical has to do with the *predisposition of each partner to collaborate in a reciprocal manner*. We call *commitment* this predisposition to collaborate and be helpful, provided that the other will reciprocate (Frank, 1988; Trivers, 1971). This concept comes from a long tradition in sociology and distributed AI, and it seems to offer a solid ground for conceptualizing emotions (Becker, 1960; Dongha, 1994; Fikes, 1982; Frank, 1988; Gasser, 1991; Gerson, 1976; Gouldner, 1960; Jennings, 1993; Kerr & Kaufman-Gilliland, 1994; Kiesler, 1971). Yet we put this concept to special use in our model (Aubé, 1997, 1998, 2001; Aubé & Senteni, 1995, 1996a, b). We call these commitments *second-order resources* to distinguish them from simpler resources such as food, water, or shelter, and we contend that, in nurturing species such as birds or mammals, they are often as critical as first-order resources. For instance, newborns and offspring in these species certainly would not survive if they could not get hold of them: through the operation of attachment structures, which rely heavily on emotions, they have to monitor and manage the disposition of their parents to take care of them. In highly social species such as with humans, commitments are essential. In a similar way that needs could be seen as computational systems that handle the management of first-order resources, we propose that emotions are computational systems that handle the management of second-order resources (or commitments) that are at the root of sociality (Castelfranchi, 1990). The challenge of mastering these powerful resources likely generated a strong selective pressure for species that could profit from collaboration, and by the same token, *a critical requirement for a new layer of design* (Krebs, 1987; Nesse, 2001).

This view of commitment explains why anger is so universally triggered by transgression. Such behavior reveals a breach in commitments and a risk of losing the second-order resources they encapsulate. Moreover, these matters have to be settled through communication, because actual modifications in commitments could only come from an exchange between the partners involved. Signalling the breach, alerting about the risks raised by the breach, and even threatening if the stake is high, is precisely what being angry is about. In the case of negative emotions, such as anger, sadness, or guilt, where there is a loss of resources involved, *the experience also has to be painful*, for the same protective reason that hurting one's body has to be alerting. Reciprocally, positive emotions, such as joy, gratitude, or pride, are triggered when there are opportunities for creating new commitments (that is new resources) or for strengthening old ones. They are felt as pleasurable, for the same reason that eating, drinking, or sleeping are pleasurable, because they likewise signal gain and replenishment.

So as to obtain a better grasp of the difference between needs and emotions, compare a woman finding a beautiful and valuable jewel, with the same woman receiving the same jewel as a gift from a loved one. *Pleasure* is felt in the first case, because a first-order resource is gained. But *joy* is felt in addition to pleasure in the second case, because of the strengthening of an important and valuable commitment. Something similar (though less profound) also happens when you take your dog for a walk, as opposed to letting him wander alone along the same path. And a very deep joy is clearly expressed by the animal in the first case. Now, think about the effect of this joy upon you, and especially upon your relationship with your dog. Shared joy creates and strengthens commitments between partners. Sadness, on the other hand, is a negative emotion raised when there is a significant loss of resources. Although it is usually considered as the opposite of joy, it nevertheless operates subtly and efficiently so as to maintain and enforce commitments. By signalling a breach in them, as when we have lost close people or kin, it also gives the measure of the resources at stake in making us suffer a great deal for it. Sadness also acts as a call to the resources of commitments by bringing closer together friends and allies. It is so efficient that it even works with animals. For example, if you forgot to feed your dog at dinnertime, he will probably solve his hunger problem by whining, and his dependence on you to satisfy his hunger need will likely activate your obligation to take good care of him. Children all over the world and offspring of many species regularly resort to emotions to fulfil their needs, in a similar fashion.

Guilt also appears notably as a commitment protective device. It is triggered whenever one feels that he or she has not fulfilled a promise or behaved properly or respected a social standard. A typical consequence is an urge to make excuses and amendments or reparation. Some researchers have questioned the adaptive character of this emotion, because empirical data clearly reveal that people typically feel greater guilt *when they do not intend* to misbehave or to be harmful to others (Frijda, 1993; McGraw, 1987). This is clearly what a commitment model such as ours would predict. If someone has been mean to you and you seek revenge, you will probably not feel much remorse in acting out your plans against that person. But if you accidentally stumble against an old woman in a rush to your office, you might feel quite bad. In the first case, there is no commitment broken on your part, and you are feeling anger yourself. In the second case, you broke a shared standard by being irresponsibly dangerous in public places, against a vulnerable person who had done absolutely no harm to you. The painful feeling serves as a warning to be more cautious in the future.

Empirical data also reveal that the closer the victim is to the person responsible for misbehavior, the more intense is the guilt felt (Baumeister, Stillwell, & Heatherton, 1994). In consonance with our model, the commitment threatened is also much higher in that case, because people generally invest more resources and confidence in close friends and kin.

The commitment model operates as a kind of magnifying glass that reveals the inner structure and the dynamics of emotions. Of course, it may happen that the first-order resources expectedly associated with a commitment are not delivered, or some of the tasks required to get them are not fulfilled, for reasons that are not under the control of one party. If the situation is clear for both partners, they might not consider the commitment to be broken, and in this case, anger will not be triggered. But for all emotions, there seems to be a *built-in bias* in the direction of overprotecting the commitments. This is the cost to be paid in the perilous world of cooperation and interactions. In the case of a loss, this is a bias of suspicion against treachery and defection (Barkow, Cosmides, & Tooby, 1992; Cosmides, 1989; Pinker, 1997). Even animals will tend to look for a scapegoat, preferably of the same species, when something harmful suddenly happens to them (Hutchinson, 1972). In the case of a gain, it is a bias in favor of gratitude, and it plays a great role in taming, seducing, and seeking new alliances (Tesser, Gatewood, & Driver, 1968).

Our analysis leads us to envision commitment as a critical concept for the understanding of emotions. To proceed further, we have to be more specific as to the nature and composition of commitment. We already stated some of the essential components of commitments. First, commitments involve *two or more partners*. They are also about *access to first-order resources*, although these might be loosely specified, as in the obligation for parents to feed their children. These resources could already be available, but often, there are certain *tasks to fulfil* to obtain them. Clear-cut *conditions of satisfaction and revocation* also constitute an integral part of commitments. Finally, commitments involve *time schedules* within which they should be satisfactorily fulfilled. Contrary to popular belief, no commitment lasts forever, and the ones that seem to endure correspond with many shorter ones that have been frequently reassessed. This is why love relations often come to an end if they are not regularly replenished, why they could start anew with someone else, and also why older children can progressively take a distance from their parents as they build up new family relations on their own. It should be clear that we do not consider love as an emotion but rather as a rich texture of densely woven commitments. As such, they offer plenty of opportunities for emotions of all kinds to be triggered, whenever any of them is at stake. *Love is an incubator of emotions*.

Commitments have a definite structure that is summarized in Table 1. They are resources, but not static ones. They behave as dynamic entities or *demons*

Table 1. Basic structure of commitments

- Commitments are *dynamic entities*;
- Commitments representation include:
    - the *partners* between which they have been settled,
    - the *first-order resources* they guarantee access to,
    - the *tasks* that are to be executed for their fulfilment,
    - the *time schedule* for their fulfilment,
    - their *conditions of satisfaction or revocation*;
- Commitments constantly *watch for events* that may bear impact upon them;
- Commitments determine the *elicitation structure* of emotions;
- Commitments are created, strengthened or protected *by the operation of emotions*
- Commitments are basically modified through *emotional acts* of expression

running in parallel in the background and watching for events that could strengthen or threaten them. This should not be too surprising, because it is similar to the way resources are dynamically represented in other motivational systems. In the case of hunger, for instance, the level of glucose circulating through the blood flow provides a reasonable assessment of the nutriments available from minute to minute.

Some of the components evoked above are combined into an *elicitation structure* that triggers the appropriate emotions depending on the commitment concerned (Aubé, 1998, 2001; Roseman, Aliki Antoniou, & Jose, 1996). It takes into account whether there is a gain or a loss incurred (*valence*), what is the importance of the variation detected (*intensity*), whether it is actual or merely anticipated (*certainty*), and which partner is seen as responsible for it (*agency*). This generally suffices to specify the proper emotion to be called. A gain in commitments attributable to someone else will thus result in gratitude being launched, while the uncertain risk of a loss, without any agent being clearly identified as responsible, will likely evoke fear as a request for help. Whenever a commitment is seriously compromised, a call is made to the appropriate emotion, and *emotional acts* of expression are intensely used to patch the breach or to exploit the new opportunity.

In closing this section, it should be stressed that such an analysis does not aim at reducing the foremost importance of needs and other reactive systems in a model of the mind. It stresses that a new level of requirements had to be met with the advent of commitments, and that specific kinds of design had to be implemented to handle the more complex behaviors that resulted from their emergence. As suggested by Norman (1980), we believe that emotions form kind of an intermediate layer of design in between reactive processes and more reflective ones, and that they probably even contributed to the emergence of the more complex structures that subsequently developed on top of them.

Critical Consequences of the Model

In the beginning of the chapter, we recalled that seeing perceptual systems as energy transducers was fruitful and promising for understanding their nature and functions as well as for implementing them in artificial systems. In a similar way, we have looked for principles that could help us see emotions in a generic fashion, so they can be more easily conceptualized and implemented in artifacts. In this endeavour, we were preoccupied that our model be largely consonant with extant data from the cognitive psychology of emotions and with principles from the theory of evolution. We submit here that the idea of envisioning emotions as commitments handlers meets the challenge. In this section, we examine some of the critical consequences this view might have on the overall structure of the mind, and, more practically, the consequences this view might have on implementation matters.

We suspect that the emergence of emotions as commitments operators had dramatic consequences on the evolution of our mental capacities. We mentioned that emotions managed to play their part by resorting to communicative acts. Although much simpler than speech acts that they preceded in evolution by some 200 millions years, emotions operate as powerful devices for conveying one's intentions to other individuals of the same species. They even cross the species barrier for animals of similar ascendance, as seems to be the case between humans and other animals such as dogs, cats, or chimps. Considering the importance speech acts hold in pragmatics and in the underlying structure of human language, we are tempted to think that emotions may have paved the way to the development of these linguistic capacities in higher primates.

We also see in *emotions-as-commitments-handlers* one of the basic roots for identity. Psychologists such as George Herbert Mead (1934) suggested that the self emerges from social interactions, when the individual gets to see himself and his own role from the point of view of the others, and when he incorporates this external stance within his own. The management of commitments, which emerged as a prerequisite for attachment behavior, also requires that individuals be differentiated from each other as unique identifiable partners. It usually is not in the best interest of animal parents that they confuse their offspring with those of others and invest too much in nurturing them. It is also not desirable that baby animals get attached to their predators. Hence, the structure of commitments management calls for the emergence of identities. Together with the communicative aspect of emotions and their likely role in the development of language, emotions as identity generators also make for good candidates at the roots of collaborative behaviors and sociality.

Moreover, if one is to cheerily protect his own commitments to others, he has to reflect upon the consequences of his own behavior upon these commitments. We have already seen that the concept of transgression is deeply ingrained in emotions such as guilt and anger, and it is also the case with others, like shame, remorse, or contempt. Emotions also make us empathetic to the fate of our fellow humans, and we often feel distress at the sight of suffering others (Levenson & Ruef, 1992; Sagi & Hoffman, 1976). We also specified earlier that the structure of commitments had to incorporate some kind of watching *demons* responsible for monitoring conduct and insuring that promises, norms, standards, or any other commitment to which we adhere are kept secure and protected. Thus envisioned, emotions appear at the root of moral behavior and consciousness (De Wall, 1996; Tappolet, 2000).

Finally, such an analysis bears some radical consequences upon the implementation of artifacts that are eventually to have emotions. We mentioned many times that emotions resemble needs, but we also specified on what grounds the two of them could unequivocally be distinguished from each other. We do not think that such a clear-cut distinction between these motivational systems has been proposed in any other model. This analysis enables us to distinguish between different categories of behaviors that answer to different sets of requirements and are thus implemented through different levels of designs, but which are unfortunately merged together in various theories. Fear, as we mentioned, is an example. For instance, we think that what makes a fish or an amphibian flee when you get too close is simpler than what makes a frightened kid call for his parents, or what makes an adult give an alarm call to allies in the presence of danger. We would suggest that the fish fear belongs to the need level. Our model postulates that these different reactions, which are too often lumped under the same emotional label, likely belong to different control mechanisms with different evolutionary histories, and that they are computed by different brain circuits. Even in humans, animal phobias, for instance, are probably much more primitive than social phobias. They respond to different kinds of drugs, and Öhman (1986, 1993) has suggested that they stem from an older *predatory-defence system*, while social fears result from a *dominance-submissiveness system* which appeared more recently in evolution. Such a difference calls for different designs and involves the management of different resources that could not be detected and handled by simpler devices.

Robotics engineers know fairly well how to implement the motivational control structures of needs within their artifacts (Maes, 1991; Steels & Brooks, 1995). We have said that the regulatory structure of emotions was similar to that depicted in Figure 2, except that it had to handle resources of a more complex nature, namely, commitments. Then, we listed in Table 1 some of the specifications that seem mandatory to represent and handle these second-order resources. Our analysis also commands that commitments be clearly represented within emotional robots as dynamic computational identities, perhaps as *actors* (Agha, 1986), that could detect and monitor events that could threaten their integrity or that could strengthen it. Finally, we sketched the triggering structure for the elicitation of individual emotions. This elicitation mechanism has to be detailed further, especially in terms of incorporating various templates that could be used to accelerate the detection of critical situations. Some of these will certainly have to do with recognizing emotional expressions, others will have to do with detecting threatening objects or animals, and clearly, many more are still to be specified. Finally, one essential consequence of our model is that artificial emotions would only make sense for communities of artifacts behaving in closely interacting groups, wherein each individual has its own recognizable identity. Provisions for constructing such an identity interactively would also have to be implemented in artifacts for them to become truly emotional, and this will not be a simple task. Members of such a community should also be built along a similar *ontology*, so they would be compelled to react similarly to the same emotional signals, and so they could not escape from being moved whenever some of their second-order resources are perceived as suddenly threatened or unusually favored.

Conclusion

In this chapter, we resorted to a requirement-based analysis so as to sketch the design of emotional systems. This led us to clearly identify emotions as motivations and to compare them with other motivational systems, such as needs. All such systems appeared as powerful control structures dedicated to the management of resources critical for survival. The basic distinction between these systems appeared to rely upon the kinds of resources each is designed to handle. We then made a thorough analysis of the communicative aspect of emotions, of the antecedent situations that typically trigger them across cultures and species, and of the neural structures that are responsible for their operation. All this led us to think that the critical resources emotions are designed to handle have to do with affective bonding and social cooperation. Yet, we had to realize that these resources do not rest upon the other agents but rather upon their mental disposition to be reliably helpful and caring. We called this new kind of resource, *commitment*, which individuals of social species have to manage so as to benefit from the help of others without incurring the risk of being neglected by indifferent relatives or paying the cost of being exploited by profiteers. Parent–offspring attachment could be seen as the prototype of such commitments, and some researchers go so far as to suggest that this basic relation likely provides the paradigm scenario of all successive collaborative behaviors. We showed that this concept was helpful in unfolding the dynamics of many emotional episodes, and we tried to specify some of the characteristics and components of commitments. It became clear that they are to be seen as dynamic entities, capable of detecting signals or events that could bear impact upon them, either threatening them or strengthening them. Emotions emerge from our analysis as the beautiful computational designs that have been carved along evolution to manage the critical resources commitments represent. We also hinted that the advent of these new requirements and designs likely had dramatic consequences on the evolution of mammals, and especially of higher primates. By resorting to communicative acts so as to modify commitments and operate upon them, emotions might well have paved the road to our complex linguistic capabilities. By having to precisely identify the partners involved in the fulfillment of commitments, they also favored the emergence of a sense of identity, which might have been transformed into a sense of self at the root of our consciousness. Finally, by vigilantly protecting all the commitments woven in our social life, emotions might as well have laid the foundation of our sense of responsibility and of complex moral behaviors. One indication pointing in that direction is that sociopaths, who typically exhibit irresponsible and amoral behavior, are also characterized by an inability to form lasting commitments and by a marked deficit of emotions (Mealey, 1995).

References

Agha, G. (1986). *Actors*. Cambridge, MA: MIT Press.

Ainsworth, M. D. S. (1989). Attachments beyond infancy. *American Psychologist, 44*, 709–716.

Aronson, E. (1996). *The social animal* (7th ed.). New York: W. H. Freeman.

Aubé, M. (1997). Toward computational models of motivation: A much needed foundation for social sciences and education. *Journal of Artificial Intelligence in Education, 8*(1), 43–75.

Aubé, M. (1998). A commitment theory of emotions. In D. Canamero (Ed.), *Emotional and intelligent: The tangled knot of cognition. Papers from* *the 1998 AAAI Fall Symposium* (pp. 13–18). Menlo Park, CA: AAAI Press.

Aubé, M. (2001). From Toda's urge theory to the commitment theory of emotions. *Grounding emotions in adaptive systems*. Special issue of *Cybernetics and Systems: An International Journal, 32*(6), 585–610.

Aubé, M., & Senteni, A. (1995). A foundation for commitments as resource management in multi-agents systems. In T. Finin, & J. Mayfield (Eds.), *Proceedings of the CIKM Workshop on Intelligent Information Agents*. Baltimore, MD.

Aubé, M., & Senteni, A. (1996a). Emotions as commitments operators: A foundation for control structure in multi-agents systems. In W. Van de Velde, & J. W. Perram (Eds.), *Agents breaking away, Proceedings of the Seventh European Workshop on MAAMAW, Lecture Notes on Artificial Intelligence, No. 1038* (pp. 13–25). Berlin: Springer.

Aubé, M., & Senteni, A. (1996b). What are emotions for? Commitments management and regulation within animals/animats encounters. In P. Maes, M. Mataric, J. -A. Meyer, J. Pollack, & S. W. Wilson (Eds.), *From Animals to Animats 4: Proceedings of the Fourth International Conference on Simulation of Adaptive Behavior* (pp. 264–271). Cambridge, MA: The MIT Press/Bradford Books.

Austin, J. L. (1962). *How to do things with words*. Cambridge, MA: Harvard University Press.

Averill, J. R. (1979). Anger. In H. E. Howe, & R. A. Dienstbier (Eds.), *Nebraska Symposium on Motivation 1978: Vol. 26* (pp. 1–80). Lincoln, NE: University of Nebraska Press.

Averill, J. R. (1983). Studies on anger and agression. Implications for theories of emotion. *American Psychologist, 38*, 1145–1160.

Axelrod, R. (1984). *The evolution of cooperation*. New York: Basic Books.

Barkow, J. H., Cosmides, L., & Tooby, J. (1992). *The adapted mind: Evolutionary psychology and the generation of culture*. Oxford: Oxford University Press.

Baumeister, R. F., & Leary, M. R. (1995). The need to belong: Desire for interpersonal attachments as a fundamental human motivation. *Psychological Bulletin, 117*(3), 497–529.

Baumeister, R. F., Stillwell, A. M., & Heatherton, T. F. (1994). Guilt: An interpersonal approach. *Psychological Bulletin, 115*(2), 243–267.

Beaudoin, L. P., & Sloman, A. (1993). A study of motive processing and attention. In A. Sloman, D. Hogg, G. Humphreys, A. Ramsay, & D.

Partridge (Eds.), *Prospects for artificial intelligence — Proceedings of AISB'93* (pp. 229–234). Oxford: IOS Press.

Becker, H. S. (1960). Notes on the concept of commitment. *American Journal of Sociology, 66*, 32–40.

Bloch, S. (1989). Émotion ressentie, émotion recréée. *Science et Vie, Hors série, 168*, 68–75.

Bloch, S., Orthous, P., & Santibanez-H, G. (1987). Effector patterns of basic emotions: A psychophysiological method for training actors. *Journal of Social and Biological Structures, 10*, 1–19.

Boucher, J. D. (1983). Antecedents to emotions across cultures. In S. H. Irvine, & J. W. Berry (Eds.), *Human assessment and cultural factors* (pp. 407–420). New York: Plenum.

Bowlby, J. (1969). *Attachment and loss. Vol. 1: Attachment*. London: Penguin Books.

Bowlby, J. (1973). *Attachment and loss. Vol. 2: Separation, anxiety, and anger*. London: Penguin Books.

Bowlby, J. (1980). *Attachment and loss. Vol. 3: Loss, sadness, and depression*. London: Penguin Books.

Castelfranchi, C. (1990). Social power. A point missed in multi-agent DAI and HCI. In Y. Demazeau, & J. -P. Müller (Eds.), *Decentralized A.I.* (pp. 49–62), London: Elsevier Science Publishers.

Cialdini, R. B. (1984). *Influence*. New York: William Morrow and Company.

Cohen, P. R., & Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence, 42*, 213–261.

Cohen, P. R., & Perrault, C. R. (1979). Elements of a plan-based theory of speech acts. *Cognitive Science, 3*, 177–212.

Cohen, P. R., Morgan, J., & Pollack, M. E. (Eds.). (1990). *Intentions in communications*. Cambridge, MA: MIT Press.

Cosmides, L. (1989). The logic of social exchange: Has natural selection shaped how humans reason? Studies with the Wason selection task. *Cognition, 31,* 187–276.

Damasio, A. R. (1994). *Descartes' error: Emotion, reason and the human brain*. New York: Avon Books.

Damasio, A. R. (1999). *The feeling of what happens: Body and emotion in the making of consciousness*. New York: Harcourt Brace and Company.

Damasio, A. R. (2003). *Looking for Spinoza: Joy, sorrow, and the feeling brain*. New York: Harcourt Brace and Company.

De Wall, F. (1996). *Good natured. The origins of right and wrong in humans and other animals*. Cambridge, MA: Harvard University Press.

Dimberg, U. (1982). Facial reactions to facial expressions. *Psychophysiology, 19*(6), 643–647.

Dongha, P. (1994). Toward a formal model of commitment for resource bounded agents. In M. J. Wooldridge, & N. R. Jennings (Eds.), *Intelligent agents, Proceedings of the ECAI-94 Workshop on Agent Theories, Architectures, and Languages, Lecture Notes on Artificial Intelligence, No. 890* (pp. 86–101). Berlin: Springer-Verlag.

Eibl-Eibesfeldt, I. (1975). *Ethology: The biology of behavior* (2nd ed.). New York: Holt, Rinehart and Winston.

Ekman, P. (Ed.) (1982). *Emotion in the human face*. Cambridge: Cambridge University Press.

Ekman, P. (1984). Expression and the nature of emotion. In K. R. Scherer, & P. Ekman (Eds.), *Approaches to emotion* (pp. 319–343). Hillsdale, NJ: Lawrence Erlbaum Associates.

Ekman, P. (1993). Facial expression and emotion. *American Psychologist, 48*, 384–392.

Ekman, P., & Davidson, R. J. (Eds.). (1994). *The nature of emotion: Fundamental questions*. Oxford: Oxford University Press.

Ekman, P., Levenson, R. W., & Friesen, W. V. (1983). Autonomous nervous system activity distinguishes between emotions. *Science, 221*, 1208–1210.

Evans, D. (2002). *Emotion: The science of sentiment*. Oxford: Oxford University Press.

Fikes, R. E. (1982). A commitment-based framework for describing informal cooperative work. *Cognitive Science, 6*, 331–347.

Frank, R. H. (1988). *Passions within reason: The strategic role of the emotions*. New York: W. W. Norton and Company.

Frijda, N. H. (1989). The functions of emotional expression. In J. P. Forgas, & J. M. Innes (Eds.), *Recent advances in social psychology: An international perspective* (pp. 205–217). Amsterdam: North-Holland.

Frijda, N. H. (1993). The place of appraisal in emotion. *Cognition and Emotion, 7*(3/4), 357–387.

Gasser, L. (1991). Social conceptions of knowledge and action: DAI foundations and open systems semantics. *Artificial Intelligence, 47*, 107–138.

Gerson, E. H. (1976). On "quality of life." *American Sociological Review, 41*, 793–806.

Gouldner, A. W. (1960). The norm of reciprocity: A preliminary statement. *American Sociological Review, 25*(2), 161–178.

Hatfield, E., Cacioppo, J. T., & Rapson, R. L. (1994). *Emotional contagion*. Cambridge: Cambridge University Press.

Hoffman, H. S. (1974). Fear-mediated processes in the context of imprinting. In M. Lewis, & L. A. Rosenblum (Eds.), *The origins of fear* (pp. 25–48). New York: John Wiley and Sons.

Hutchinson, R. R. (1972). The environmental causes of agression. In J. K. Coles, & D. D. Jenson (Eds.), *Nebraska Symposium on Motivation 1972: Vol. 20* (pp. 155–181). Lincoln, NE: University of Nebraska Press.

Jennings, N. R. (1993). Commitments and conventions: The foundation of coordination in multi-agent systems. *Knowledge Engineering Review, 8*, 223–250.

Johnson-Laird, P. N., & Oatley, K. (1989). The language of emotions: An analysis of a semantic field. *Cognition and Emotion, 3*, 81–123.

Kerr, N. L., & Kaufman-Gilliland, C. M. (1994). Communication, commitment, and cooperation in social dilemmas. *Journal of Personality and Social Psychology, 66*(3), 513–529.

Kiesler, C. A. (1971). *The psychology of commitment: Experiments linking behavior to belief*. New York: Academic Press.

Klinnert, M. D., Campos, J. J., Sorce, J. F., Emde, R. N., & Svejda, M. (1983). Emotions as behavior regulators: Social referencing in infancy. In R. Plutchik, & H. Kellerman (Eds.), *Emotion: Theory, research, and experience. Vol. 2. Emotions in early development* (pp. 57–86). New York: Academic Press.

Krebs, D. (1987). The challenge of altruism in biology and psychology. In C. Crawford, M. Smith, & D. Krebs (Eds.), *Sociobiology and psychology: Ideas, issues and applications* (pp. 81–118). Hillsdale, NJ: Lawrence Erlbaum Associates.

Larsen, R. J., Kasimatis, M., & Frey, K. (1992). Facilitating the furrowed brow: A nonobtrusive test of the facial feedback hypothesis applied to unpleasant effect. *Cognition and Emotion, 6*(5), 321–338.

LeDoux, J. (1998). *The emotional brain: The mysterious underpinnings of emotional life*. New York: Simon and Schuster.

Lemerise, E. A., & Dodge, A. D. (1993). The development of anger and hostile interactions. In M. Lewis, & J. M. Haviland (Eds.), *Handbook of emotions* (pp. 537–546). New York: Guilford Press.

Levenson, R. W., & Ruef, A. M. (1992) Empathy: A physiological substrate. *Journal of Personality and Social Psychology, 63*(2), 234–246.

Lewis, M. (1993). Self-conscious emotions: Embarrassment, pride, shame, and guilt. In M. Lewis, & J. M. Haviland (Eds.), *Handbook of emotions* (pp. 563–573). New York: Guilford Press.

Lindsay, P. H., & Norman, D. A. (1972). *Human information processing: An introduction to psychology*. New York: Academic Press.

Lutz, C. (1988). Ethnographic perspectives on the emotion lexicon. In V. Hamilton, G. H. Bower, & N. H. Frijda (Eds.), *Cognitive perspectives on emotion and motivation* (pp. 399–419). Dordrecht: Kluwer.

Lutz, T. (2001). *Crying: The natural and cultural history of tears*. New York: W. W. Norton and Company.

MacLean, P. D. (1993). Cerebral evolution of emotion. In M. Lewis, & J. M. Haviland (Eds.), *Handbook of emotions* (pp. 67–83). New York: Guilford Press.

Maes, P. (Ed.). (1991). *Designing autonomous agents: Theory and practice from biology to engineering and back*. Cambridge, MA: MIT Press.

Malatesta, C. Z., & Haviland, J. M. (1982). Learning display rules: The socialization of emotion expression in infancy. *Child Development, 53*, 991–1003.

McGraw, K. M. (1987). Guilt following transgression: An attribution of responsibility approach. *Journal of Personality and Social Psychology, 53*(2), 247–256.

Mead, G. H. (1934). *Mind, self, and society from the standpoint of a social behaviorist*. Chicago, IL: University of Chicago Press.

Mealey, L. (1995). The sociobiology of sociopathy: An integrated evolutionary model. *Behavioral and Brain Sciences, 18*, 523–599.

Minsky, M. (1985). *The society of mind*. New York: Simon and Schuster.

Nesse, R. M. (Ed.). (2001). *Evolution and the capacity for commitment*. New York: Russell Sage Press.

Norman, D. A. (1980). Twelve issues for cognitive science. *Cognitive Science, 4*, 1–32.

Oatley, K., & Johnson-Laird, P. N. (1996). The communicative theory of emotions: Empirical tests, mental models, and implications for social interaction. In L. L. Martin, & A. Tesser (Eds.), *Striving and feeling. Interactions among goals, affect and self-regulation* (pp. 363–393). Hillsdale, NJ: Lawrence Erlbaum Associates.

Öhman, A. (1986). Face the beast and fear the face: Animal and social fears as prototypes for evolutionary analyses of emotion. *Psychophysiology, 23*, 123–145.

Öhman, A. (1993). Fear and anxiety as emotional phenomena: Clinical phenomenology, evolutionary perspectives, and information-processing mechanisms. In M. Lewis, & J. M. Haviland (Eds.), *Handbook of emotions* (pp. 511–536). New York: Guilford Press.

Ortony, A., Clore, G. L., & Collins, A. (1988). *The cognitive structure of emotions*. Cambridge: Cambridge University Press.

Panksepp, J. (1991). Affective neuroscience: A conceptual framework for the neurobiological study of emotions. In K. T. Strongman (Ed.), *International review of studies on emotion: Vol. 1* (pp. 59–99). New York: John Wiley & Sons.

Pinker, S. (1997). *How the mind works*. New York: W. W. Norton and Company.

Provine, R. R. (2000). *Laughter: A scientific investigation*. New York: Penguin Books.

Roseman, I. J., Aliki Antoniou, A., & Jose, P. E. (1996). Appraisal determinants of emotions: Constructing a more accurate and comprehensive theory. *Cognition and Emotion, 10*(3), 241–277.

Rosenblum, L. A., & Alpert, S. (1974). Fear of strangers and specificity of attachment in monkeys. In M. Lewis, & L. A. Rosenblum (Eds.), *The origins of fear* (pp. 165–193). New York: John Wiley & Sons.

Sagi, A., & Hoffman, M. L. (1976). Empathic distress in the newborn. *Developmental Psychology, 12*(2), 175–176.

Scherer, K. R. (Ed.). (1988). *Facets of emotion: Recent research*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Scherer, K. R., Wallbott, H. G., & Summerfield, A. B. (Eds.). (1986). *Experiencing emotion: A cross-cultural study*. Cambridge: Cambridge University Press.

Searle, J. R. (1969). *Speech acts: An essay in the philosophy of language*. Cambridge: Cambridge University Press.

Searle, J. R. (1979). *Expression and meaning: Studies in the theory of speech acts*. Cambridge: Cambridge University Press.

Simon, H. (1967). Motivational and emotional controls of cognition. *Psychological Review, 74*, 29–39.

Sloman, A. (1987). Motives, mechanisms, and emotions. *Cognition and Emotion, 1*(3), 217–233.

Sloman, A. (1995, April). *Architectures for emotional agents*. Conference presented at the Geneva Emotion Week, Université de Genève, Geneva.

Steels, L., & Brooks, R. (Eds.). (1995). *The artificial life route to artificial intelligence*. Hillsdale, NJ: Lawrence Erlbaum Associates.

Stein, N. L., Trabasso, T., & Liwag, M. (1993). The representation and organization of emotional experience: Unfolding the emotion episode. In M. Lewis, & J. M. Haviland (Eds.), *Handbook of emotions* (pp. 279–300). New York: Guilford Press.

Strack, F., Martin, L. L., & Stepper, S. (1988). Inhibiting and facilitating conditions of the human smile: A nonobtrusive test of the facial feedback hypothesis. *Journal of Personality and Social Psychology, 54*(5), 768–777.

Tangney, J. P. (1995). Shame and guilt in interpersonal relationships. In J. P. Tangney, & K. W. Fischer (Eds.), *Self-conscious emotions. The psychology of shame, guilt, embarrassment, and pride* (pp. 114–139). New York: Guilford Press.

Tappolet, C. (2000). *Émotions et valeurs*. Paris: PUF.

Tesser, A., Gatewood, R., & Driver, M. (1968). Some determinants of gratitude. *Journal of Personality and Social Psychology, 9*(3), 233–236.

Toates, F. (1986). *Motivational systems*. Cambridge: Cambridge University Press.

Trivers, R. L. (1971). The evolution of reciprocal altruism. *The Quarterly Review of Biology, 46*, 35–57.

Vallerand, R. J., & Thill, E. E. (1993). Introduction au concept de motivation. In R. J. Vallerand, & E. E. Thill (Eds.), *Introduction à la psychologie de la motivation* (pp. 3–39). Laval, Québec: Éditions Études Vivantes.

Winograd, T., & Flores, F. (1986). *Understanding computers and cognition: A new foundation for design.* New York: Addison Wesley.

Chapter 3

Metaphor, Self-Reflection, and the Nature of Mind

John A. Barnden
University of Birmingham, UK

Abstract

This chapter speculatively addresses the nature and effects of metaphorical views that a mind can intermittently use in thinking about itself and other minds, such as the view of mind as a physical space in which ideas have physical locations. Although such views are subjective, it is argued in this chapter that they are nevertheless part of the real nature of the conscious and unconscious mind. In particular, it is conjectured that if a mind entertains a particular (metaphorical) view at a given time, then this activity could of itself cause that mind to become more similar in the short term to how it is portrayed by the view. Hence, the views are, to an extent, self-fulfilling prophecies. In these ways, metaphorical self-reflection, even when distorting and inaccurate, is speculatively an important aspect of the true nature of mind. The chapter also outlines a theoretical approach and related implemented system (ATT-Meta) that were designed for the understanding of metaphorical discourse but that incorporate principles that could be at the core of metaphorical self-reflection in people or future artificial agents.

Introduction: What Questions Are We Addressing?

(a) What is mind?

(b) What are theories of mind?

(c) What could or should computationally implemented architectures and systems based on theories of mind be like?

(d) How should we respond to a particular sort of fragmentation in the study of mind?

In this chapter, these questions are asked from the indirect point of view of *how a mind views itself or other minds* rather than directly from the theoretical observer's point of view of determining what the mind *really* is. As a result, reflection on the above issues (a through d) is roughly as follows, where (a) and (b) have been collapsed together:

(ab2) How does a mind view itself (what kinds of theories does it have about itself); how does it view other minds; and how do these matters interact with the question of what minds really are?

(c2) What could or should computationally implemented architectures and systems involving minds' views of minds be like?

(d2) How should we respond to a particular sort of fragmentation in minds' views of minds?

The move to these issues from Issues a through d might be thought to be twisting the latter too far. But behind *ab2* is a claim that *how a mind views itself is part of, and can affect, the real nature of that mind itself.* After all, one important aspect of a mind is its process of thinking (consciously or unconsciously) about itself. *How* it views itself is then a bald fact about that mind. For instance, to make the point vividly, it may view itself as being a physical being trapped inside the body and able to have a life outside that body if only it could get out. Theories of mind must take into account the views and theories that minds have about themselves and each other, even if they are highly inaccurate or irrational. If a mind thought it was made out of fire and water, then the fact that it thought that is an important fact about that mind, even though it is not actually made out of fire and water.

But, more deeply, the author will claim that a view that a mind has of its own nature at some particular time can, so to speak, entrain that mind to become, at around that time, more in accordance with that view than it would have been without the operation of the view. Views of oneself can, to some extent, become self-fulfilling prophecies.

The main ideas in this chapter concern an individual mind's intermittent views of itself, rather than of other minds, and are thus consonant with the need to include self-reflection capabilities in architectures of complete minds. However, a view that a mind has of itself at some moment could be influenced by the views it perceives other minds as intermittently having of themselves and each other. Views of mind, as with views of anything else, can be transmitted from person to person.

Issue (d), as opposed to (d2), is about disciplinary fragmentation in research on mind. The findings in this chapter implicitly contribute to curing (d) to some extent, even though it discusses (d2) instead, and is indeed in the direction of supporting the idea that the mind has a natural tendency to have a fragmented overall view of itself. The chapter material is disciplinarily integrative in bringing a computational outlook to bear on deep philosophical and psychological issues and naturally supports a link between the study of language about the mind and the study of mind; in particular, metaphor is given a central place in the study of mind. Moreover, "mind" is taken to include affect. The way a mind views its own affect is important here, as are the affective aspects of the way a mind views itself (even the nonaffective features of itself).

The plan of the chapter is as follows. Section 2 provides background on metaphor and its relationship to thought and the study of thought. Section 3 is the start of the main thrust of the chapter and argues that for reasons of practical necessity, self-reflection is likely to be importantly metaphorical. Section 4 discusses ways in which metaphoricity of self-reflection could distort the true nature of the mind as opposed to merely presenting a distorted picture of a mind to itself. Section 5 presents a case for some qualia in consciousness to be intrinsically metaphorical in nature. Section 6 discusses the fragmentary nature of self-reflection that is likely to arise from metaphor, but which may be unavoidable in any case. Section 7 briefly outlines the author's approach to the understanding of metaphorical discourse, and an implemented system (ATT-Meta) derived from it, and shows how the approach could provide principles and techniques that could be at the core of a metaphorically self-reflective agent. Section 8 concludes.

Throughout the chapter, it is important for the reader to bear in mind that the main concern is with what views a mind might take of its own internal states and processes in the short term for some particular cognitive purpose, rather than with views a mind might continuously have about itself in the long term. Of least concern is the question of how similar those views are, or should be, to attempted objective accounts of the mind that might be devised by scientists or philosophers:

in other words, accounts that minds might have of themselves as a result of extended intellectual deliberation as opposed to resulting from the ordinary life experience that anyone could have without being a scientist or philosopher.

Background: Metaphor, Discourse, Mind, and Affect

One aspect of a complete mind, situated within anything like our world, must be the ability to reason about other minds and about itself as a complete mind. Now, as cognitive linguists and others have shown (see, for example, Lakoff, 1993), much human discourse concerning minds is highly metaphorical. Some particular, common ways in which discourse talks metaphorically about mind are as follows.

- **Mind as Physical Space.** We commonly talk about minds as if they were physical containers or physical regions. This is typified by utterances such as "The idea invaded my mind," "She pushed the idea to the back of her mind," "The fear was buried deep within his mind," and "In the far reaches of her mind, she knew that her husband had been unfaithful."
- **Ideas as Living Creatures.** We commonly talk of ideas as if they were living creatures, as in "The belief had been lurking in her mind," "Several different desires were battling inside her," "The thought of his impending arrival was tugging insistently at her," and one of the examples above of *Mind as Physical Space*: "The idea invaded my mind."
- **Ideas as Physical Objects.** The metaphorical view of *Ideas as Living Creatures* is a special case of a more general, pervasive metaphorical view of ideas as physical objects that could be inert. This appears in utterances such as two of the examples above of *Mind as Physical Space:* "She pushed the idea to the back of her mind" and "The fear was buried deep within his mind;" as well as in utterances such as "The suspicion stuck to her like a magnet," "They kicked ideas around the room," and "She was still a long way from a solution," where the ideas are in an external physical space (either a real one surrounding the person mentioned or an imaginary physical space outside the person).
- **Cognition as Perception.** We often talk of cognition as if it were, or included, physical perception, as in "She couldn't focus clearly on the problem," "He could picture it very clearly in his mind," "The memory was

lost in the mists of her mind," and "The situation stank of corruption." The visual case, as in the first three of these examples, is particularly common.

- **Mind Parts as Persons.** A metaphorical view that is less often remarked upon, but that is nevertheless commonplace, and plays a significant role in this chapter, is where a mind is viewed as being made up of or containing subpeople, with their own thoughts, desires, etc., and possibly communicating with each other, as in "Part of him was afraid of raising the issue," "Part of me could see that my sister had been dishonest," "One part of me was whispering that I ought to leave, while another part was begging me to stay," and "The child inside him was crying for attention." Locutions such as "being in two minds about *X*" should perhaps be classified here as well, together with some of the metaphors discussed by Lakoff (1996).

Other metaphorical views of mind, together with abundant examples, can be found in many sources, including the author's own databank at http://www.cs.bham.ac.uk/~jab/ATT-Meta/Databank, which also has links to other sites. (We use the term "metaphorical view" to mean roughly what Lakoff and others mean by a "conceptual metaphor"—essentially, a mapping from aspects of one domain to supposedly corresponding aspects of another.)

Discourse can switch rapidly between different, possibly conflicting views in describing mental states and processes. Discourse often mixes different views together, as will be illustrated below and as is already exemplified by the fact that some of the utterance examples listed above are included under more than one view. This is all in perfectly mundane discourse, not (just) poetry and other literary art, as our examples illustrate.

A further tenet held by many metaphor researchers (for example, Gibbs, 1994; Lakoff 1993) is that the metaphorical views used in discourse are, generally speaking, crucial aids in thought (conscious or unconscious) rather than just linguistic decoration. In this chapter, this is assumed to be true. Thus, our starting point is that people's conscious and unconscious thinking, not just their discourse, about each other and about themselves is partly and perhaps highly metaphorical. This is not to say that minds *believe* that the views are true, they are just useful ways of thinking, consciously or unconsciously. We also assume that people can adopt different views at different times and in different circumstances, and can switch rapidly between different views, even when thinking about one person's thoughts, just as in discourse.

The hypothesis that people use metaphorical views not only in thinking about other people's minds but also in thinking about their own is suggested by the observation that metaphorical talk about mental states is often in the first person. In the case of *Mind Parts as Persons*, for instance, spoken uses of the view are often, and perhaps mostly, in the first person, although third-person uses are common in novels, etc. As for other metaphorical views, examples such as the following are common in ordinary discourse:

- It was in the back of my mind.
- The thought crept into my mind.
- The thought stuck to me.
- I said to myself that ...
- My mind felt totally focused.

We will assume in this chapter that there are several possible ways in which a particular person can be led to use a particular metaphorical view in thought or language. People may, possibly, be genetically predisposed to think of minds in a particular way. They may develop particular ways as a result of observing the ways their own minds work and considering how other people's minds may be working. Or, they may learn to use a view because they encounter it frequently in discourse. Clearly, therefore, the views a particular person uses may be strongly influenced by the language and culture that that person is embedded in and by societal norms laid down in that culture about how people should think of themselves or others (see, for example, Johnson, 1985). Another cultural aspect is that much of religious belief is about the nature of the self. The degree of variety, therefore, in a particular person's use of metaphor of mind may be partly dependent on culture; however, an added complication is that the way metaphor enters into the person's unconscious thoughts may be different from the way it enters into their conscious thoughts, and from the point of view of the present discussion, unconscious thoughts are of great potential importance.

In natural language discourse, affective states are often described metaphorically (see, for example, Fainsilber & Ortony, 1987; Kövecses, 2000). Examples are "His anger boiled over" and "Sweet feelings welled up within him." A mind's internal reflection on its own affect can therefore be conjectured to involve metaphor. Also, metaphor is often used in natural language discourse to convey value judgments and emotions about the targeted subject matter (often to deviously smuggle them in, but also, more beneficently, to convey them in an economical and effective way). For instance, hearing poverty being described as a disease could cause one to have particular negative emotions about poverty or poor people. Similarly, *thinking* of poverty as a disease could have such effects. Thus, we may conjecture that a mind's affective states can, in part, be caused by that mind's metaphorical thoughts as well as being explicitly described in that mind's metaphorical thoughts.

We assume also that a metaphorical view that someone takes of some entity can affect not just the person's reasoning, emotions, value judgments, and communication about it but, therefore, also how the person deals with it (interacts with it, manipulates it, controls it, etc.). For example, thinking of poverty as a disease can affect one's attempts to control it or can affect one's interactions with poor people. This effect is at the root of the use of metaphor in political discourse in order to persuade people to adopt particular stances or behaviors (see, for example, Mio, 1997).

Discussions of self-reflection, except when conducted by a metaphor researcher, rarely engage with the way that metaphor might enter into self-reflection. This is one example of the fragmentation of research emphasized by Issue (d). We must note carefully that many discussions of consciousness refer to or use metaphorical notions of mind, such as the Cartesian Theatre and internal narratives (Dennett, 1991), the mind's eye (Rorty, 1980), and global workspaces (Baars, 1993). However, what is predominantly at issue here is the question of what metaphors it is appropriate for an observing theoretician to use or to avoid in elucidating the true nature of consciousness, *not* with the question of how the use of such metaphors within the self-reflection of the observed mind can affect the nature and functioning of that mind. One type of exception to this neglect is the study within the area of psychiatric therapy of how people's metaphors about their own selves affect their moods and their thoughts about themselves (Mio & Katz, 1996). Of course, the popular self-help literature is full of concepts such as "inner child" and "playing negative tapes in one's head," together with instructions about how to attend to, exploit, or avoid such metaphors in controlling one's inner and outer lives, although the metaphoricity may not be explicitly recognized in a particular tract.

We take the views discussed in this section to be metaphorical, even though there may be senses in which some of them could be construed as literal. For instance, if one holds the philosophical view that ideas are neural patterns of activation and allows that such a pattern is a physical object, then ideas really are physical objects. They would then also have particular physical locations inside the head, whether in the sense of being physically confined to one small region of the brain or in the looser sense of being spread over many widely distributed, but nevertheless specific, neurons in the brain. Furthermore, the ideas could change location. However, we are concerned with the *Mind as a Physical Object* view as part of the common sense of an ordinary person, and as part of philosophically and scientifically uninformed discourse, not dependent on knowledge of how the brain works or of theories about the relationship between mind and body. Another example is that one might propose that the mind is in fact made up of subagents that possess separate beliefs, desires, etc. However, even if it is, discourse involving *Mind Parts as Persons* is not dependent on it being the case or on any suspicion by the participants that it is the case, and that metaphorical view may conflict with the discourse participants' scientific theories, if any, about the nature of mind. A further observation about *Mind Parts as Persons* is that while it is sometimes used with an implication that the mind parts in question are permanent features of that mind, as in "My inner child is always feeling jealous so my adult self spends a lot of time arguing with it," it is also often used with no such implication, as in "One part of me can see the force of that argument," where the part is not further characterized in the discourse as having any special, long-term existence or qualities.

Metaphorical Self-Reflection as a Practical Necessity

In the previous section, metaphorical self-reflection was mentioned as a mere special case of self-reflection in general. However, it is plausible to suggest that there may, in fact, be no practical alternative to doing substantial amounts of reasoning about mental states by means of metaphor.

First, as can be seen from the literature on metaphor, such as the work by Lakoff (1993), it is plausible that there is no practical alternative to metaphor for thinking about messy abstract domains, especially when matters are complex or subtle. It is widely acknowledged that metaphor, when applied appropriately to messy domains, can provide more economical and precise description, and more effective reasoning, than is otherwise practical. It may not even be possible to describe some things without metaphor (see, for example, Stern, 2000, on the problem of unparaphrasability of much metaphor). As an example of the difficulty of doing away with metaphorical description, it is difficult to paraphrase the sentence "In the murky depths of her mind, Anne realized that her husband had been unfaithful" without resorting to other metaphors that capture ways in which Anne's thoughts can be relatively inaccessible to her. At least, it is difficult to do without resorting to lengthy circumlocution.

Second, it is plausible that one's own mind is a messy, complex, and subtle domain for oneself as well as for others, even if one has some sort of privileged, direct access to one's own mind. The potential messiness, complexity, and subtlety is, if anything, increased by having more extensive access to one's own mental states than to those of other minds.

Furthermore, an agent *X* can be expected to learn metaphorical ways of talking and thinking about minds, from the metaphorical ways that other agents use in their speech. These ways of talking about minds could be absorbed by *X* and become ways in which *X* thinks about itself. This does not, of course, preclude

X developing metaphorical and other ways of thinking about itself purely through self-reflection.

Distorting Oneself Through Metaphor

We pointed out above that a metaphorical view used in a mind's self-reflection is a real feature of that mind, no matter how inaccurate the view is. Metaphorical self-views are aspects of the real nature of mind. But, we can also see ways in which the use of a metaphorical self-view at some time can cause a mind to become, at around that time, more similar to its own view of itself than it would otherwise have been. In other words, views of oneself can become self-fulfilling prophecies, to some degree. We will now look at two possible ways in which this could happen.

Distortion Method 1: Through Metaphorical Self-Management

Any given broad metaphorical view of mental states and processes, such as *Mind as Physical Space* or *Ideas as Physical Objects,* captures some real aspects of mind and ignores others. This is just a special case of a general feature of metaphorical description—different metaphorical views generally capture different aspects of what is being viewed (Grady, 1997; Lakoff & Johnson, 1980). Moreover, even when a view captures certain features, it generally does so only approximately. For example, a view of marriage as a journey approximately captures ways in which the relationship can develop, whereas a view of marriage as a business contract approximately captures ways in which the partners interact or should interact with each other at any moment in time.

Thus, if a mind's self-management is partially influenced at some point in time by a particular metaphorical view *V*, the self-management may be partially defective because of the inaccuracies of *V*. But, the operations of self-management may, to some extent, tend to make the mind behave as if it were more accurately described by *V*. We can call this phenomenon the distortion of oneself through metaphorical self-management.

A vivid commonplace example of this potential effect is when people view themselves as having an "inner child" partially governing their thoughts and actions. To the extent that they believe that children should not be overly controlled, or cannot be controlled, they may refrain from taking self-control actions that they would otherwise take (and be perfectly able to take). Similarly, to the extent that they believe a child should be controlled or influenced by certain methods, they may take analogous self-control actions. In these ways, they would come to act and think more as if they really had a child inside controlling things.

One general point here is that if a metaphorical view used in a mind's self-reflection fails to be sensitive to particular opportunities for external or internal actions by that mind, self-management may be deprived of the opportunity for exploiting those possibilities, so that the mind does not perform actions that it could, in fact, perform. To take a less vivid example than that of the inner-child, a certain person *Z* may at some point in time be viewing her own current mental operations via the metaphorical view of *Mind as Physical Space*. Some of her self-management could be influenced by her perceptions of how "central" some ideas are in her mind-space (that is, her perceptions of how strongly she is attending to them), and could be conducted with the aim of "moving" ideas closer or further from the centre. She might assume that bigger "movements" require more effort, so that less central ideas require more time and effort. She might therefore attend even less to ideas she perceives as being on the periphery of her mind, even if, unbeknownst to her, they do not require significantly more time or effort to deal with. Those ideas could then become less attended to than they were already. Thus, the metaphorical view *Z* is taking of herself can tend to cause her to exaggerate certain features of her mental state that are (inaccurately) captured by the view.

This example is related to the more general observation that people can become limited by the views that they hold about themselves. Someone who believes he cannot do something will tend not to try to do it, whatever it is, and may also engage in behavior precluding it. Through such effects, the person may become less able in some capacity than they otherwise would have been.

The example above was about *Z* refraining from certain mental acts. More positively, a metaphorical self-view could lead *Z* to engage in certain types of mental behavior that she would otherwise have been less likely to engage in. For example, consider the common metaphorical view of *Ideas as Internal Utterances*, used in sentences like "One part of her whispered that she was wrong." Suppose that at some moment in time *Z* is experiencing and thus viewing some of her own thoughts as internal utterances. This may lead her to respond to those thoughts by further acts of mental utterance, when she might otherwise have responded by, say, constructing a mental picture or diagram. Thus, the view that happened to be in play has encouraged its own strengthening and continuance (for some period of time, which may be very short).

Distortion Method 2: Through Metaphorical Self-Conformity

The previous subsection can be roughly (and metaphorically) summed as follows: a self-reflective Part A of a mind views a Part B metaphorically and thereby distorts B (for example, by exaggerating certain qualities of it). The present subsection is, instead, about a Part B distorting itself by conforming to some view that a Part A is using in its actions toward B.

To some extent, people tend to adapt to—in the sense of coming to conform to—views that other people have of them or ways other people have of dealing with them. For example, if person B thinks A thinks B is stupid, B can start to act more stupidly than he would otherwise. Another case is illustrated by a situation where a person B acts in a businesslike way in dealings with A, because A is acting in a businesslike way with B. Quite apart from such questions as B's thinking that he *should* act in a businesslike way, in order, say, to impress or outwit A, there is simply the effect that A's businesslike dealing with B sets up a context of action where some types of action are more appropriate than others, and B may simply slide naturally into that context through a type of imitation.

Similarly, assuming that metaphorical views that people entertain about each other can affect the way they behave toward each other, it follows that someone may tend to conform, temporarily at least, to some view that is affecting the way someone else is dealing with him. B may find that A is viewing B's argumentation as physical attack and is, therefore, engaging in conversational maneuvers modelled on combat situations, thereby leading B to act similarly.

Now, could this type of metaphor-conformity effect apply also within a single mind? Suppose a mind can sometimes be legitimately viewed as being composed of two or more subagents, with mind-like capabilities and the ability to perform operations on each other, to reason about each other and to communicate with each other. That is, suppose that to some degree the mind really is, perhaps intermittently, organized according to the view of *Mind Parts as Persons.* (Even if the human mind is not normally, or ever, like this, it could be the way an artificial agent is organized.) Then is it too fanciful to suppose that a subagent B could conform to the way it is being dealt with by another subagent A, and in particular, come to behave more in accordance with some metaphorical view that A is entertaining about B? Note here that previous parts of this chapter lead to the conjecture that subagents would engage in metaphorical thought about each other just as much as a single unitary mind would engage in metaphorical self-reflection. For example, subagents might view each other as engaged in physical combat.

If such a metaphor-conformity effect could happen, it would be another way in which the overall agent is distorting itself to conform to its own metaphorical self-reflection.

Metaphorical Qualia

Suppose Bill works at the Foreign Office within the government of a hypothetical country and is metaphorically viewing the Office as a solar system, with the Foreign Secretary as the sun and junior ministers as planets. Suppose even that Bill is one of the planets. Then it could perfectly well be that Bill does not *feel* like a planet going round a sun, even partially. For example, he may well have no feeling of being, literally, physically pulled toward the Foreign Secretary, physically circling round him or her, or receiving life-giving radiation from him or her. It could well be that Bill has worked out, or learned from others, that there is a formal correspondence between certain abstract relationships and activities within the Office (or within organizations of that type in general) and the relationships and processes in a solar system.

However, it is in fact possible that Bill has some feelings that are similar to the putative feelings mentioned in the previous paragraph. For example, it could be that the feeling of loyalty toward the Foreign Secretary has something in common with feeling physically pulled toward him or her, and, more strongly, that feelings of pleasure and comfort arising from being in his or her good books could be similar to, if not actually the same as, some of the feelings of pleasure or comfort arising from basking in sunlight. Such possibilities would fit well with theories that metaphor derives partially from embodied experience (see, for example, Johnson, 1987). Equally, as many commentators have observed, talk of being hot with anger may derive in part from feelings of being hot when angry. Thus, at least in the case of some metaphors, the conscious qualia involved in the target-domain situation being described (Foreign Office, say) may be, in part, similar to or the same as some of the qualia involved in the source domain.

When applied to metaphors of mind, such considerations lead to an important additional line of thought that extends the suggestions in previous sections of this chapter. It is developed more extensively than here in Barnden (1997). We start with the observation made in an earlier section that first-person manifestations of metaphors of mind are common. The central conjecture of the present section is that we do not use such language just for practical convenience, i.e., merely because it supports useful reasoning about the described mental states, but also because, at least to a limited extent and for some of the time, it reflects the *feel* of mental states to us. Here the word "feel" has a sense as broad as the word "qualia." (Thus, in the intended broad sense, redness has a feel.) Using this broad sense, the conjecture is that, for instance:

- Thought can (sometimes) *feel* like internal speech.
- Thought can (sometimes) *feel* like vision.
- One's mind can (sometimes) *feel* like a physical space, and one can *feel* that one's ideas are far apart or moving around within that space, or coming into the space from outside.
- One may (sometimes) *feel* that inside of one there are several independent thinking entities with their own thoughts and feelings.

Now, if this is the case, then, because these feelings are part of the conscious mind, it follows that at least some metaphorical views of mind are, in part, intrinsic aspects of the nature of the consciousness, not just convenient tools for describing mental states. This is a new way, going beyond the points made in previous sections, in which metaphor-based self-reflection is part of the *actual* nature of mind and not just a matter of inaccurate views of the actual nature of mind.

The case of cognition feeling like vision and other types of perception is especially interesting, as it connects to the study of mental imagery in psychology and philosophy and to old debates about the role of imagery in cognition and consciousness (see Glasgow, 1993, for a review and an artificial intelligence model). To liken, say, conscious visual imagery to an activity of picturing is to say that conscious visual imagery feels, to some extent, like seeing a picture. Notice that this feeling is a real part of the person's current state of consciousness, even if it is epiphenomenal in the sense of it not having any effect on the person's mental processes. Also, see Horne (1993) on the sensuous nature of imagery.

One recent theory that relates cognition strongly to perception is that of Barsalou (1999). Barsalou downplays the role of metaphor, but the mental use of perception-based metaphors for abstract concepts, including concepts about the mind, is not antithetical to his claims. His theory, in relying heavily on simulation of perceptual processes and on the stimulation of related affective states, would then tend to support the idea that thoughts couched in terms of the source domain (for example, the domain of physical objects) in a metaphor for mind would make the person experience the qualia that would arise in that source domain.

Fragmentation of a Mind's Overall View of Itself

As we saw earlier, a given metaphorical view captures only part of the targeted phenomenon, and different views capture different parts (in general). Also, because of inaccuracies in the capturing performed by different views, the views can conflict in what they convey about the target. A business metaphor for marriage emphasizes divergence of goals, whereas a journey metaphor emphasizes commonality of goals.

Thus, it is natural to expect that, on the assumption that self-reflection in minds is importantly metaphorical, there will necessarily be an important degree of fragmentation and inconsistency in self-reflection, and these effects stand to be heightened by the potentially self-fulfilling nature of metaphorical views. This may sound like a disadvantage. However, given a need for self-views to be importantly metaphorical, it is an advantage for there to be multiple views: they can potentially offset each other or be applicable in different situations, providing, overall, a higher degree of accuracy and completeness of self-reflection than would arise from using any individual view.

The mentioned fragmentation and inconsistency are primarily an observation about human minds. However, metaphor could provide to artificial minds a useful tool for description of mental states, capturing complexities, subtleties, and messiness that it would otherwise be difficult to deal with. Thus, metaphorical self-reflection and management could be useful. But this would bring in fragmentation and inconsistency. This chapter proposes that this outcome should simply be embraced. After all, nonmetaphorical self-reflection would also probably have to involve oversimplifications and, therefore, inaccuracies, so there might need to be multiple, partially inconsistent self-views even if they were all nonmetaphorical.

Toward the Future: The ATT-Meta System and Approach

The author developed a theoretical approach and an implemented artificial intelligence system, called ATT-Meta, for conducting metaphor-based reasoning (Barnden, 1998, 2001; Barnden et al., 1994; Lee & Barnden, 2001). This has been applied largely to the special case of metaphor-based reasoning about mental states. For example, it can trace through implications of two ideas being "far apart" in a mind considered as a physical region. The intended ultimate purpose of the methods used in the system is for them to form part of natural language discourse processing. However, the techniques used in the system could also be used reflectively by a mind to reason about itself on the basis of metaphorical self-reflection. In this section, we comment on some features of the system and the underlying theoretical approach, in the spirit of indicating how the types of mental processing discussed in previous sections could realistically form part of a mind design.

In the ATT-Meta approach, the understanding agent (or an agent using metaphor in its own thought processes) is assumed already to have acquired knowledge of a range of commonly used metaphorical views. Recall that a metaphorical view is essentially a mapping of aspects of the source domain (for example, physical space) to aspects of the target domain (for example, mind). We assume that the individual mapping relationships making up the mapping are general in nature. For example, the ATT-Meta system's knowledge of the *Ideas as Physical Objects* view is largely a matter of a mapping physical manipulation (of ideas that are being viewed as physical objects) to mental usage of those ideas, with no specific types of manipulation, such as banging or sawing, being mapped. We also assume that the view maps physical interaction between an agent's ideas to conjoint mental usage of those ideas by the agent. Similarly, the *Mind as Physical Space* view is largely a matter of mapping physical presence in the space to existence in the mind in question.

The approach powerfully exploits such mapping relationships by allowing for an indefinite amount of reasoning within the terms of the source domain. For instance, suppose an utterance says that two ideas are "far apart" in someone's mind. We assume the understander takes this to be portraying the ideas as physical objects that are physically far apart in that mind conceived of as a physical space (so the utterance relies both on *Ideas as Physical Objects* and on *Mind as Physical Space*). Given the common-sense source-domain knowledge that physical objects do not normally interact to any substantial degree when far apart (at least in the everyday physical world), we get the source-domain inference that the two mentioned ideas are probably not physically interacting to any substantial degree. Therefore, via the mapping relationship mentioned above, we get the target-domain conclusion that the ideas are probably not being used conjointly to any substantial degree by the agent in question (so, for example, the agent will not infer consequences of the two ideas taken together).

The source-domain reasoning can be much richer than this. For instance, if an idea is being portrayed as being in the murky depths of someone's mind, knowledge of how murk and physical depth can affect physical visibility, accessibility, and manipulability will be used to connect to general mapping relationships of the type illustrated above. (In this example, the view of *Cognizing as Seeing* comes into play as well as the two views used above.) A distinctive feature of our approach to metaphor compared to most others is its allowance for an indefinite amount of complex source-domain reasoning; exceptions include the approaches of Hobbs (1990) and Narayanan (1997). One thing unique to ATT-Meta, however, is an implemented, tested system that allows the source-domain reasoning to be arbitrarily interleaved with other reasoning operations, such as target-domain reasoning and mapping operations between source and target. It is to be expected that this interleaving would be important for a realistic application of the approach to the self-reflection concerns of the present discussion.

The ATT-Meta approach has as one of its basic principles that of *Map-Extension Minimization.* This can be illustrated with the murky-depths example. The approach avoids, if it can, trying to map the murky depths themselves over to the target domain. Rather, it is only the mappable *effects* in source-domain terms that are mapped over: in this case, the effects of being inaccessible, etc. The reason for adopting this stance is that in many cases there is simply not enough nonmetaphorical knowledge about how minds work to be able to find target-domain correspondents for things like murky depths, and we suspect that such concepts are used in metaphorical utterances purely for the effects they have. Another reason for the stance is that it can be extremely expensive in computational terms to search for a coherent partial isomorphism between two domains (Falkenhainer et al., 1989). In any case, we would claim that in many cases there simply is no isomorphism to be found or intended by the speaker.

Also distinctive in the approach is a worked-out and implemented account of how the compounding of metaphorical views works. This is hinted at in the above examples, as there were two or three views compounded. Our approach to compounding (which we also call mixing, though without the negative connotation that the term "mixed metaphor" is often taken to have) is discussed further in Lee and Barnden (2001). One feature of our approach is attention to the distinction between parallel mixing (where a domain is viewed at the same time in terms of several source domains, as in the above examples) and serial mixing (usually called chaining, where Domain A is viewed in terms of Domain B, which is in turn viewed in terms of a Domain C). A (real) example of serial mixing is "The thought of my mother-in-law's arrival hung over me like an angry cloud," where the thought is viewed as a cloud, and the cloud is viewed as an agent that has an emotion. Given that metaphor compounding is not rare in ordinary discourse about mind, it is reasonable to take it as something that would need to be accounted for in metaphorical self-reflection.

An additional feature of the approach is that it allows for graded effects, in two senses. First, there is a handling of uncertainty and of conflict between lines of reasoning. This is important in view of the fact that common-sense knowledge and reasoning are typically uncertain (for example, it is only usually the case that physical objects that are far apart do not interact very much), and the outputs of metaphorical mapping may conflict with target-domain knowledge and reasoning, which may be uncertain. The other graded effect is that things can be the case to varying degrees: for example, objects that are close together can interact to a high degree, whereas objects that are far apart interact only to a low degree. Notice that such degrees are orthogonal to the question of uncertainty. Some low degree of physical interaction may be supposed to exist with high certainty, and a high degree of interaction may be supposed with low certainty.

As regards the nature of source-target mapping, we have so far only mentioned mapping relationships that are specific to particular metaphorical views, such as the mapping relationship from physical manipulation to mental usage. However, our approach also incorporates view-neutral mapping adjuncts (VNMAs), which are mapping principles that apply by default whatever the particular metaphorical views are in operation (Barnden & Lee, 2001). For example, the temporal order of events in the source domain is assumed to map to give the same ordering of any corresponding events in the target domain. Another example of a VNMA is that the degree of ease with which something can be done in the source domain maps to the degree of ease of corresponding actions (if any) in the target domain. As illustrated in Barnden (2001b), it is often the case that many of the important effects of a metaphorical utterance occur via VNMAs rather than via the view-specific mapping relationships, which may merely supply a substrate of correspondence on which VNMAs then work. A few VNMAs have been realized in our implemented system, but considerable work remains to be done on implementing them.

From the point of view of the present chapter, a particularly important VNMA is one that transfers information about emotion from source to target. Specifically, an aspect of the VNMA is that the metaphorically reasoning agent's own emotions about a source-domain situation are also assumed by default to be emotions it has about the corresponding target-domain situations (if any). For example, if the self-reflecting mind is saddened by something expressed in source-domain terms, then it is (probably) saddened by the real situation portrayed.

As explained in Barnden, Glasbey, and Wallington (to appear) and Barnden, Glasbey, Lee, and Wallington (2004), the ATT-Meta approach makes some major claims about discourse-extended metaphor. The approach claims that it is a mistake to assume that the metaphor side of the task of understanding discourse is to convert each metaphorical utterance into nonmetaphorical internal-representational terms. It is often better, more economical and more effective, for the understander to keep thinking in terms of the source domain over the course of understanding several utterances, and only cross over to the target domain when there is a real need (for example, when information needs to be integrated with information conveyed by nonmetaphorical utterances). In this way, some individual utterances may merely contribute to an overall source-domain scenario out of which selected aspects are mapped (aspects that may have no simple relationship to any one utterance), rather than contribute target-domain information individually. We should expect that in metaphorical self-reflection it could be useful to pursue the self-reflection over a considerable period of time in source-domain terms, only crossing over into the target domain when necessary. In short, not every episode of metaphorical self-reflection need have direct target-domain consequences of its own.

Closely related to these ideas is our argument (Barnden, Glasbey, Lee, & Wallington, 2004) that it is useful to be able to transfer information from target to source as well as in the usual direction of source to target. For example, when a metaphorical view is extended over a stretch of discourse, information derived from interspersed nonmetaphorical utterances can beneficially be converted into the terms of the prevailing metaphorical view(s). This allows integration of information to happen in source-domain terms rather than in target-domains terms. We argue that integration on the source side is often easier and richer than integration on the target side. Correspondingly, in metaphorical self-reflection, it could be beneficial to engage in target-to-source mapping to achieve integrated thinking about oneself.

Finally, the ATT-Meta system also has facilities for reasoning uncertainly in nonmetaphorical terms about agents' beliefs and reasoning. It allows for any degree of nesting (reasoning about agents reasoning about agents reasoning about …). Thus, the approach is relevant to arbitrarily nested self-reflection. Metaphor can appear anywhere in the nesting.

Conclusion

We addressed the question of the views that minds can have of themselves, rather than directly addressing the question of what minds are really like. However, we noticed that these views are nevertheless part of the real nature of mind and may additionally be an intrinsic aspect of conscious qualia. Moreover, we conjectured that entertaining a particular view of itself for a particular cognitive purpose can cause a mind to become more similar at around that time to how it is portrayed by the view. This could happen through at least two different mechanisms: roughly speaking, one aspect of the mind could distort another by acting on it in conformity with a metaphorical view, or one aspect could distort itself by coming more into conformity with a view of it employed by another aspect.

Although these points could be argued to apply to any sort of view, we concentrated on the special case of metaphorical views. Metaphorical views may be needed in practical self-reflection, just as they are needed in practical natural language discourse about mind, because of the messiness, complexity, and subtlety of mental states and processes. Metaphorical views throw into especially sharp relief the likely partiality and inaccuracy of individual views and the inconsistency between different views.

Our work on the ATT-Meta system for metaphorical reasoning provides ideas on how natural and artificial minds could think metaphorically, and thus make the general considerations of this chapter more real for mind researchers.

Acknowledgments

This research was supported by grant GR/M64208 from the Engineering and Physical Sciences Research Council of the United Kingdom. This chapter was developed from a talk created for Aaron Sloman's symposium *How to Design a Functioning Mind* at the 2000 Convention of the Society for Artificial Intelligence and Simulation of Behaviour (AISB-00), held at the University of Birmingham, UK. The talk was given in adapted form at the EURESCO conference *Mind, Language and Metaphor: Euroconference on Consciousness and the Imagination*, Kerkrade, The Netherlands, in April 2002.

References

Baars, B. J. (1993). Why volition is a foundation problem for psychology. *Consciousness and Cognition, 2*(4), 281–309.

Barnden, J. A. (1997). Consciousness and common-sense metaphors of mind. In S. O'Nuallain, P. McKevitt, & E. Mac Aogain (Eds.), *Two sciences of mind: Readings in cognitive science and consciousness* (pp. 311–340). Amsterdam/Philadelphia: John Benjamins.

Barnden, J. A. (1998). Combining uncertain belief reasoning and uncertain metaphor-based reasoning. In *Proceedings of the 20th Annual Meeting of the Cognitive Science Society* (pp. 114–119). Mahwah, NJ: Lawrence Erlbaum Associates.

Barnden, J. A. (2001a). Uncertainty and conflict handling in the ATT-Meta context-based system for metaphorical reasoning. In V. Akman, P. Bouquet, R. Thomason, & R. A. Young (Eds.), *Proceedings of the Third International Conference on Modeling and Using Context* (pp. 15–29). Lecture Notes in Artificial Intelligence, Vol. 2116. Berlin: Springer.

Barnden, J. A. (2001b). *Application of the ATT-Meta metaphor-understanding approach to selected examples from Goatly.* Technical Report CSRP-01-01, School of Computer Science, The University of Birmingham, UK.

Barnden, J. A., Glasbey, S. R., Lee, M. G., & Wallington, A. M. (2004). Varieties and directions of inter-domain influence in metaphor. *Metaphor and Symbol, 19*(1), 1–30.

Barnden, J. A., Glasbey, S. R., & Wallington, A. M. (to appear). Metaphor and truth from an artificial intelligence standpoint. In A. Burkhardt, & B. Nerlich (Eds.), *Reflections on tropical truth: Studies on the epistemology of metaphor.*

Barnden, J. A., Helmreich, S., Iverson, E., & Stein, G. C. (1994). An integrated implementation of simulative, uncertain and metaphorical reasoning about mental states. In J. Doyle, E. Sandewall, & P. Torasso (Eds.), *Principles of knowledge representation and reasoning: Proceedings of the Fourth International Conference* (pp. 27–38). San Mateo, CA: Morgan Kaufmann.

Barnden, J. A., & Lee, M. G. (2001). *Understanding open-ended usages of familiar conceptual metaphors: An approach and artificial intelligence system.* Technical Report CSRP-01-05, School of Computer Science, The University of Birmingham, UK.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22,* 577–660.

Dennett, D. C. (1991). *Consciousness explained.* London: Penguin.

Fainsilber, L., & Ortony, A. (1987). Metaphorical uses of language in the expression of emotions. *Metaphor and Symbolic Activity, 2*(4), 239–250.

Falkenhainer, B., Forbus, K. D., & Gentner, D. (1989). The Structure-Mapping Engine: algorithm and examples. *Artificial Intelligence, 41*(1), 1–63.

Gibbs, R. W., Jr. (1994). *Poetics of mind: Figurative thought, language and understanding.* Cambridge: Cambridge University Press.

Glasgow, J. I. (1993). The imagery debate revisited: A computational perspective. *Computational Intelligence, 9*(4), 309–333.

Grady, J. E. (1997). THEORIES ARE BUILDINGS revisited. *Cognitive Linguistics, 8*(4), 267–290.

Hobbs, J. R. (1990). *Literature and cognition.* Stanford University, CA: CSLI Press.

Horne, P. V. (1993). The nature of imagery. *Consciousness and Cognition, 2*(1), 58–82.

Johnson, F. (1985). The western concept of self. In A. J. Marsella, G. DeVos, & F. L. K. Hsu (Eds.), *Culture and self: Asian and Western perspectives* (pp. 91–138). London: Tavistock.

Johnson, M. (1987). *The body in the mind.* Chicago, IL: Chicago University Press.

Kövecses, Z. (2000). *Metaphor and emotion: Language, culture, and body in human feeling.* Cambridge: Cambridge University Press.

Lakoff, G. (1993). The contemporary theory of metaphor. In A. Ortony (Ed.), *Metaphor and thought* (2nd ed.). Cambridge: Cambridge University Press.

Lakoff, G. (1996). Sorry, I'm Not Myself Today: The metaphor system for conceptualizing the self. In G. Fauconnier, & E. Sweetser (Eds.), *Space, worlds, and grammar* (pp. 91–123). Chicago, IL: University of Chicago Press.

Lakoff, G., & Johnson, M. (1980). *Metaphors we live by.* Chicago, IL: University of Chicago Press.

Lee, M. G., & Barnden, J. A. (2001). Reasoning about mixed metaphors with an implemented AI system. *Metaphor and Symbol, 16*(1&2), 29–42.

Mio, J. S. (1997). Metaphor and politics. *Metaphor and Symbol, 12*(2), 113–133.

Mio, J. S., & Katz, A. N. (Eds.). (1996). *Metaphor: Implications and applications.* Mahwah, NJ: Lawrence Erlbaum Associates.

Narayanan, S. (1997). *KARMA: Knowledge-based action representations for metaphor and aspect.* Ph.D. thesis, Computer Science Division, EECS Department, University of California, Berkeley, August 1997.

Rorty, R. (1980). *Philosophy and the mirror of nature.* Oxford: Blackwell; Princeton, NJ: Princeton University Press.

Stern, J. (2000). *Metaphor in context.* Cambridge, MA; London: Bradford Books, MIT Press.

Chapter 4

Modular Representations of Cognitive Phenomena in AI, Psychology, and Neuroscience

Joanna J. Bryson
University of Bath, UK

Abstract

Many architectures of mind assume some form of modularity, but what is meant by the term 'module'? This chapter creates a framework for understanding current modularity research in three subdisciplines of cognitive science: psychology, artificial intelligence (AI), and neuroscience. This framework starts from the distinction between horizontal modules that support all expressed behaviors vs. vertical modules that support individual domain-specific capacities. The framework is used to discuss innateness, automaticity, compositionality, representations, massive modularity, behavior-based and multi-agent AI systems, and correspondence to physiological neurosystems. There is also a brief discussion of the relevance of modularity to conscious experience.

Introduction

Many of the architectures of mind described and referred to in this book assume some form of modularity. But what is considered to define a module varies a great deal both within and across the cognitive science disciplines: artificial intelligence (AI), psychology, and neuroscience. This chapter is not devoted to any one architecture (though I have one too, which I will describe briefly in the *Discussion* to make my biases clear), but is rather an overview of the concepts and concerns of modularity. It covers all of the above disciplines and shows how they relate to one another and to cognition—or at least to cognitive phenomena such as planning, learning, language, emotions, and consciousness.

My hope is that this chapter will serve as a useful primer—in the best case, a Rosetta stone—for scientists and laypeople trying to get a handle on what the various fields of cognitive science might mean by modularity, and how the modular architectures described in this book and elsewhere might correspond to our common understanding of what minds do. It is important to realize that researchers who are experts in one or more of the areas described below may have no awareness of some of the other areas, and therefore may make no effort to reconcile their own theories with the others. Consequently, this chapter contains some substantial redescription in an effort to put these theories into a common framework for comparison.

Modularity in Psychology

Criteria for Modularity

I will begin with an extremely simple definition of modularity from the psychological literature, due to Flombaum et al. (2002):

Modularity is the thesis that the mind contains independent input systems that, when engaged, are restricted in the types of information that they can consult.

This definition is useful for two reasons. First, it introduces a very clean criterion for modularity: that some part of the mind does not have access to some other part of the mind, or at least not its "information." Given this simple criterion, anyone who accepts the idea of implicit knowledge or unconscious behaviour has already acknowledged that there is some sort of modularity involved in human intelligence.

This is not the only possible characterization of modularity. Fodor (1983) provided the best-known list of criteria for recognizing modularity, some of which, for example innateness, are now highly controversial. To be fair, the entire concept of innateness has become controversial, because the lifelong interplay between genetics and environment makes many (particularly postmodern) developmental psychologists uncomfortable with the category (for example, Elman et al., 1996; Thelen & Smith, 1994; and Donnai & Karmiloff-Smith, 2000). Those who do not believe in innateness as a discriminative category in human development often do not believe in modularity either, because Fodor (1983) famously staked so much importance on the innateness criteria. This seems slightly ridiculous when coming from an AI perspective, because innateness has no bearing on the functional or computational characteristics of modularity, but it has had a large impact on the psychological modularity literature.

Psychologists who believe in modularity are generally concerned with other traits, such as automaticity in the presence of appropriate stimuli or brain localization. I will discuss brain localization under neuroscience below. Automaticity is indicated both by speed of processing and by changes in processing due to nearly identical but saliently different stimuli that help the module select its own input. Modules are expected not only to be specialized to a domain but also to be able to recognize the context in which their domain is present.

An impressive example of this is shown by Tanaka and Farah (1993) in the domain of face recognition. Recognizing individuals from their faces is an extremely difficult, highly skilled behaviour that takes years to develop. Children tend to recognize people through superficial cues such as glasses and hairstyle. Adults with sufficient experience make discriminations on subtle differences between faces. Experience is critical: even adults often have trouble discriminating faces from less familiar races, while those with specialized experience, such as farmers and field researchers, can learn to discriminate the faces of other species. Nevertheless, face recognition has often been seen as a candidate module, because we do it quickly with no deliberate access to the process, and because the capacity to recognize faces can be lost through brain damage or stroke, independently of any other capacity.

Tanaka and Farah (1993) contributed to this debate by demonstrating implicit, automatic context recognition by the face recognition capacity. Starting with photographs of famous faces, they divided the pictures down the middle, and then shifted one side slightly with respect to the other. Despite the fact that the misalignment boundary was quite conspicuous—subjects were aware that they saw a face that was slightly skewed—both speed and accuracy of recognition were substantially degraded. The Fodorian modularist's explanation is that these slightly altered visual stimuli no longer trigger the "face recognition" module.

In my opinion, the most critical attribute of modularity is that individual modules support and are supported by different specialized representations (Bryson & Stein, 2001a; Bryson, 2002). Notice that this is fairly compatible with the inaccessibility definition of Flombaum et al. (2002)—process structure is heavily dependent on representational structure and content, so if we consider minds to be based on process, clearly the different processes might have difficulty accessing each other's knowledge or control (at least directly) if they are based on different representations. However, I came to this conclusion in the course of designing a development methodology for artificial intelligence, which I will discuss further below.

Fodor: Vertical and Horizontal Modules

The second reason that the Flombaum et al. (2002) quote is a useful introduction to modularity in psychology is the phrase "independent input systems." This makes clear the origins of a great deal of the theory underlying modularity in the psychological literature—Fodor's book The Modularity of Mind (1983). Although Fodor stated that he believes modularity may also exist in motor systems (p. 42), he claims ignorance of these systems and concentrates on perception. An entire school of psychological research has followed this lead (recently, Spelke, 2003; Coltheart, 1999; Downing et al., 2001), some (such as Flombaum et al., 2002) apparently unaware that Fodor's full architecture is actually symmetric with respect to sensing and action.

Fodor cited Chomsky (1980) and Gall (1825; the originator of phrenology) as his main inspirations. Dawkins (1976) and Hume (1748) also gave highly relevant discussions. But I will use Fodor as a basis for describing psychological modularity, both because of his influence in psychology and because of his relevance to modular AI.

Fodor introduced the terms "horizontal" versus "vertical" to describe two different sorts of decomposition for intelligence. For Fodor, horizontal decompositions are those that identify processes that underlie all of cognition, such as memory, attention, perception, and judgement. Vertical decompositions identify particular skills or faculties, such as mathematics, language, and metaphysics, which each have their own characteristic processes of memory, attention, and so forth (Fodor, 1983, pp. 14–21). Roughly speaking, evidence for horizontal decomposition is the extent to which, for a particular individual, performance across all domains is correlated, while evidence for vertical decomposition is the extent to which it is not. For example, it might turn out that individuals who have good memories tend to be able to remember things well across any domain; this would indicate that memory is a horizontal module. On the other hand, if how good an individual is at mathematics in no way predicts how good they are at language, then this is evidence that both mathematics and language are vertical modules.

Fodor believed that only certain parts of human intelligence are decomposed in the vertical sense; those parts are perception and action. In Fodor's system, a number of semiautonomous perceptual modules run simultaneously, giving quick, automatic analysis of the perceptual scene. Each module recognizes its own best input and effectively trumps any other module trying to process that input. The output of perception modules is in the "language of thought." This output is operated on by a horizontal reasoning system that then chooses an action. The chosen action is then presumably produced by a vertical action module, though as I have mentioned, such action-skill modules are little researched or discussed in the Fodorian modularity literature. But, we would expect such a module to take "language of thought" as input and to generate patterns of muscular control as output.

Even if Fodorian psychology research considered motor as well as perceptual modules, it would never consider the sorts of tightly coupled perception-motor modules prevalent in artificial intelligence (for example, Albus, 1997; Minsky, 1985; and Brooks, 1991b). I discuss these further below. This is because, for Fodor, the purpose of vertical modules is to reduce the complexity of the real world into a common representation used by a horizontal general-purpose reasoning system.

Massive Modularity and Evolutionary Psychology

The examples Fodor initially proposed of vertical modules (for example, language and mathematics) are far higher-level skills than most Fodorian psychological modularists currently ascribe to modules. This is because of another characteristic Fodor attributed to modules: that they are atomic. This means that, for Fodor, modules are not composed of further modules. Because language has been demonstrated to have many independent constituent parts, it does not meet this Fodorian criteria, despite being one of the modules he originally discussed the most. Fodor believed this atomicity was necessary to his vision of lightning-fast, automatic, parallel modules vying with each other to interpret the world for the general-purpose reasoning system (and, presumably, to translate the general-purpose reasoning back out into actions in the world).

Other researchers, coming particularly from evolutionary psychology, have a very different understanding of modularity (for example, Cosmides & Tooby, 1994; Evans & Zarate, 1999; Carruthers, 2003). These researchers focus primarily on understanding why humans show greater computational abilities in some cognitive domains than others. That is, problems that are computationally equivalent are easier or harder to solve depending on the domain being reasoned about. For example, people are better at reasoning about relationships when they are expressed in terms of social characteristics and obligations than when they are presented as logical abstractions. People are also more capable of doing arithmetic involving fractions if problems are expressed in terms of the currency local to their country. (This was easier to demonstrate before the British converted to a decimal system of change for the British Pound.)

Here again there is a diversity of opinion about innateness. For some researchers, the leading indication that a module exists is if nonhuman primates are shown to share the specialized capacity. For example, the recent results indicating monkeys expect equivalent compensation as their peers for performing the same task (Brosnan & de Waal, 2003) is taken as evidence for a cheater-detection module. Others consider any specialized capacity, such as face recognition described above, to be an indication of a module. They are happy to believe that modules develop or are learned. Such acquired modules could explain both the increased cognitive capacities of mature animals and their relative inflexibility—essentially a general-purpose learning substrate consolidates into regions of specialized skills and representations. Bates (1999) provided such an account of language learning, although she is not normally associated with massive modularity. *Massive modularity* is the term applied to those who believe the adult mind consists perhaps entirely of specialized (vertical) skill modules.

Modularity in Artificial Intelligence

I mentioned briefly above that Fodor's theory of modularity was strongly influenced by contemporaneous work by, for example, Chomsky (1980). Chomsky's influence extends not only into linguistics and philosophy but also into computer science, particularly in AI. Two of Chomsky's colleagues at MIT working in AI also made contemporaneous contributions to modularity research that have resulted in the widespread adoption of modularity in certain areas of AI.

Since the mid-1990s, modular approaches have dominated the development of "autonomous" AI systems, such as mobile robots or virtual reality (VR) characters (Kortenkamp et al., 1998; Bryson, 2000; Thórisson, 1999; Sengers, 1999; Hexmoor et al., 1997). These systems share with humans and other animals the characteristic of needing to be able to coordinate a large range of intricate expressed behaviors, many of which are only applicable in some of the variety of contexts the system may find itself in. These contexts include an environment that changes independently of the actions of the intelligent system and in ways the system cannot control.

This section offers a brief overview of four distinct approaches to modularity that were developed in AI in the last 20 years. (For more extensive reviews, see Bryson, 2000, 2001.)

Modules as Agents

The first well-known modular model of mind, at least described by an AI researcher, is Minsky's *Society of Mind* (1985). Although the book was published in 1985, Minsky had been working on and presenting the idea for some time before that (Doyle, 1983). Compared to Fodor's, Minsky's proposal is more substantially vertical, although it still has some horizontal elements. An individual's actions are determined by simpler individual agencies, which are effectively specialists in particular domains. Minsky's agencies *are* compositional—they exploit hierarchy for organization. For example, the agency of play is composed of agencies of block-play, doll-play, and so forth. Arbitration between agencies is also hierarchical, so the play agency competes with the eat agency for the individual's attention. Once play establishes control, the block and doll agencies compete.

Minsky's agents have both perception and action, but not memory, which is managed by a shared facility—presumably "horizontal" to Fodor, though one that is still modularly decomposed. Memory (K) agencies are interconnected with each other and with the other actor (S) agents. K agents and S agents can each activate the other type as well as others of their own type. Keeping the whole system working requires another horizontal faculty: the "B brain" that monitors the main (A) brain for internally obvious problems such as redundancy or feedback cycles.

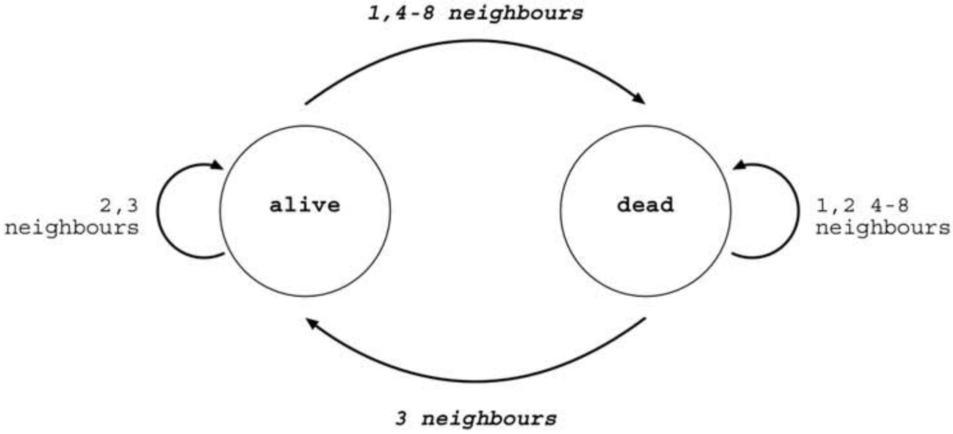
Minsky's model attempts to account for all of human intelligence but has never been fully implemented. The existent systems described in the book, for example, the learning system of Winston (1975), were for the most part fairly traditional, monolithic, single-problem AI systems with centralized control. Masters and PhD students routinely resolve to fully and properly implement the *Society of Mind* model, but there is, to date, no widely accepted canonical implementation.

Modules as Finite State Machines

In contrast, the term "behaviour-based artificial intelligence" (BBAI) was invented to describe a simplified but fully implemented system, originally used to control mobile robots. This was the subsumption architecture (Brooks, 1986, 1991b). The subsumption architecture is purely vertical. The modules were originally each finite state machines (Figure 1), and arbitration between them was conducted exclusively by wires connecting the modules, originally literally (Connell, 1990), but soon as encoded in software. Each wire could connect one module to another's input or output wires, the signal of which the first module could then either monitor, suppress, or overwrite.

Brooks initially asserted that most apparent horizontal faculties (memory, judgement, attention, reasoning) were actually abstractions "emergent from" (used to describe) an agent's expressed behaviour but had no place in the agent's actual control (Brooks, 1991b, pp. 146–147). However, his system was rapidly extended to have learning systems either inside modules or local to layers of modules (for example, Brooks, 1991a; Mataric, 1990). My opinion is that this is precisely where learning belongs, in specialized representations in the heart of modules. Unfortunately, what might have been a promising approach has generally been overlooked by most critics and followers of the subsumption architecture. They were most enthralled by the attractive and radical simplicity

Figure 1: A finite state machine is an enumerated set of all the possible states the module can be in, plus the complete list of possible transitions between states, each labelled with the condition that would lead the module to make that transition. Example figure is for the cells in Conway's Game of Life (Gardner, 1970).



of Brooks' de-emphasis of representation and centralized control. In fact, many researchers are still convinced that Brooks' robots are "stateless" (have no memory), despite the fact that finite *state* machines are at the core of his architecture and serve as the short-term memory necessary to react to events after they are sensed (for example, collisions with obstacles).

Modules as Slaves and Bitmaps

Of the researchers who did *not* immediately adopt "no representation" as a mantra, most attributed the impressive success of Brooks' approach to the fact that he had created abstracted primitives—the semiautonomous action/perception modules. Because these primitive units could sort out many of the details of a problem themselves, they made the composition of intelligence under any approach easier (Malcolm et al., 1989). Thus, modular behaviour systems have been incorporated as components into a large variety of AI architectures, many of which still maintain centralized, logic-based planning and learning systems (for example, Gat, 1991; Bonasso et al., 1997). In fact, due to the difficulty of reasoning about relatively autonomous components, some systems have reduced behaviors to "fuzzy rules" (Konolige & Myers, 1998) or vector fields (Arkin, 1998), which can be more easily composed.

Despite the lack of commonality of such approaches to Brooks' original ideal, they are still often called either behaviour-based or hybrid behaviour-based systems. Further, by the late 1990s, the work of these researchers had so far outstripped that of the "pure" BBAI researchers that two significant publications declared these hybrid approaches to have been conclusively demonstrated superior to pure BBAI (Kortenkamp et al., 1998; Hexmoor et al., 1997).

It is interesting to note that the systems with simplified, easily composed modules (for example, Konolige & Myers, 1998; Arkin, 1998) are the AI systems closest to Fodor's ideal, although often the modules are for action, not perception. But they are simple, quick, one-step mappings from a goal constructed by a centralized/horizontal planning system to a set of motor commands to achieve it. On the other hand, they have lost many of the engineering advantages that Minsky and Brooks considered critical to modular AI. Intelligence is no longer decomposed entirely into simple elements. The planning systems are generally as elaborate as any in AI; they simply reason about more powerful elements.

Agents as Modules

At the other end of the modular-complexity spectrum are multiagent systems (MASs) (Wooldridge & Ciancarini, 2001; Weiß, 1999). Here, the modules composing the system *are* agents, but not in Minsky's sense. Rather, these agents were meant, at least initially, to be complete software systems—often, the agents use the sort of hybrid behaviour-based architectures just described (Guzzoni et al., 1997; d'Inverno et al., 1997).

MAS practitioners generally consider themselves to be modelling not individual minds, but societies. They nevertheless typically have "horizontal" modules/agents/components for connecting agents with complementary needs and abilities (directory agents) or for enforcing behavioral norms of participants.

In some senses, MASs are actually closer to BBAI than the so-called hybrid behaviour-based systems. Each agent performs a particular task and may have its own private knowledge store and representations that are presumably well suited to its function. However, to date, there are a few fundamental differences between a MAS and a single, modular agent. These differences are due to issues of communication and arbitration between modules and agents. The MAS community is concerned with interoperability between unspecified numbers and types of agents, and with distribution across multiple platforms. This creates an administrative overhead not necessary for a single, modular agent. Where MASs are, in fact, limited to a single platform and a relatively fixed architecture, I suspect their engineers may be taking the wrong approach and should consider them to be modular single agents. But, this is a topic for another paper (Bryson, 2003).

It is important to realize that, despite their high profile in some research communities, MASs are not yet a proven technology (Edmonds, 2002). Unlike behaviour-based and hybrid behaviour-based systems, they do not, as yet, have an extensive commercial application base.

Summary: AI and Mental Modules

AI provides us with working models of both Fodorian modular decomposition (in the form of hybrid architectures) and of massive modularity (in the form of more strictly modular architectures, such as behaviour-based AI and multiagent systems). This provision has been largely unintentional, though certainly influenced by some concerned researchers' theories of the nature of natural intelligence. Because they have been built into working systems, they have been subjected to a special kind of selective pressure. These systems need to work, and in order to work, they need to be relatively easy to design and debug. Thus, the quest for success in AI leads to a sort of selective pressure for parsimony, yet at the same time, a need to be able to handle the complexity of the real world.

The net result of all this experience seems to be the following:

- Modularity is an important attribute for systems that have to interact with a complex, changing environment. It is used widely for mobile robotics, virtual reality, and user interfaces. It has also often been suggested for managing networked resources (whether load balancing or exploiting e-services on the Internet), but these applications are not yet well established.
- *Pure* modularity is difficult to manage. If modules are both autonomous and simple, they tend to interfere with each other. Most modular systems now have some sort of behaviour arbitration. These systems run the gamut from top-down control by a reasoning system to negotiated solutions, where each module acts as a voter. Some architects, including Gat (1998), Bryson and Stein (2001a), Blumberg (1996), and Sloman and Logan (1999), think that intermediate architectures that reflect both top-down and bottom-up information will ultimately prevail. Such architectures require additional specialized structures, akin to Fodor's horizontal modules.
- Nevertheless, extremely quick, simple modules typical of Fodor's description of vertical modules are *not* the norm, although some examples of such an approach exist. If hand-coded BBAI continues dominating applications (or is replaced by MASs), then this will be evidence that, for AI at least, it makes more sense for modules to be more intricate, mapping sensing clear through to action. To this extent, AI supports a model more like massive modularity.

For a more complete (if older) analysis along these lines, see Bryson (2000), or the slightly updated version of that work in Chapter III of Bryson (2001). I should say that although this section has concentrated mostly on the pragmatic aspects of AI, this is not meant to undermine the work by some philosophers and psychologists to work within the AI discipline at creating and understanding complete models of mind. Besides Minsky, see, in particular, Sloman and Logan (1999), as well as many of the chapter authors in this book.

Modularity in Neuroscience

We have evidence of at least three sorts of modular decomposition in mammal brains[1]: modularity by organ within the brain, by region within an organ, and by context or time. In this section, I will describe each of these in more detail.

Modularity by Organ

We know that different parts of the central nervous system have radically different structures, in terms of different component cells, different amounts of connectivity, and different organizations of connectivity. Even if we did not have behavioral evidence (as we do) that the neocortex, cerebellum, hippocampus, and so forth, perform different functions, we would suspect as materialists—and given our understanding of computation in networks—that these organs must perform different computations because of their different structures and connectivity. This point becomes even more obvious when we realize there is no particular reason not to extend the concept of organ modularity to more peripheral organs, such as the spinal cord, the retina, or the cochlea.

The brain is normally considered to have three parts: the fore-, mid-, and hindbrain (Carlson, 2000). Speaking roughly, the hindbrain seems necessary for coordinated action, as animals that are missing much of their hindbrain produce jerky, uncoordinated actions and may have difficulty with balance. The midbrain contains much of the more basic or primitive control, including the encoding of complex species-typical behaviors. A cat with an intact hind- and midbrain can go through the motions of pouncing, stretching, sleeping, and eating, but does not necessarily perform these behaviors in appropriate contexts. The forebrain is associated with connecting behaviors to contexts, or in more cognitive terms, with goal-oriented behaviour. In primates, at least, this includes inhibiting more reflexive actions so that more complex strategies or learning can have a chance to achieve activation (Hauser, 1999).

This sort of task decomposition indicates that the best parallel within the Fodorian framework to organ-level modularity may well be horizontal decomposition. Organs seem dedicated to a sort of processing, not a particular context for perception or action.

Modularity by Region

Even within an organ that is fairly structurally homogeneous (at least in considerations likely to affect the nature of its computations), there are differences in function. In some cases, these seem to be determined primarily by connectivity: for example, the primary auditory and visual cortices are areas of the neocortex that most directly receive the sensory input of the two systems. It has been suggested that other regions are modular by function, such as the "fusiform face area" or the "parahippocampal place area" (Downing et al., 2001). However, given the amazing diversity of cortical computation even in single regions (for example, Kauffman et al., 2002, show that the "visual cortex" is necessary for learning Braille, see below), it may be that such apparent specialization also reflects connectivity. For example, the fusiform face area may reflect links to subcortical brain organs specialized for purposes such as social interaction (perhaps the amygdalic system), and the "parahippocampal place area" may reflect known links to navigation in the hippocampal system.

Some cortical regions are steps along a stream of processing, for example, regions dedicated to identifying low-level features such as line orientations (Hubel, 1988) or to identifying higher-level concepts, such as categories of objects or tasks (Freedman et al., 2001) or personal identity (Perrett et al., 1992). If we are trying to map these regions into the sorts of modularity expressed by BBAI or massive modularity, we would have to consider a column of such representations as a single module or hypothesize ensembles of modules acting in a coordinated manner.

Modularity by Context

Even within a given region, the semantics of a particular cell's firing seem to be dependent on the context in which it fires. This has been demonstrated in the hippocampus (Kobayashi et al., 1997), in sensory cortices mapping receptive fields (Sen et al., 2001), and in the prefrontal cortex (Asaad et al., 2000). I believe that the extent of the consequences of this temporal modularity has not been fully recognized. It may be that some computations are mutually exclusive, because their representations cannot be active at the same time. Further, individual differences in developing these representations might account for individual differences in insight and generalization based on the relative accessibility of two representations. See, for example, Skaggs and McNaughton (1998) for their account of individual differences in rats' ability to discriminate two similar rooms.

The concept of temporal modularity is in marked contrast to the descriptions of Fodor or the rationale for BBAI and of MASs, which claim that the reason the modular approach is useful is because all modules are constantly active. However, some more established (and less conventionally modular) systems of cognitive modelling, such as Soar (Newell, 1990) and ACT-R (Anderson, 1993), have found it necessary to create *problem spaces* so that the system chooses actions only from an appropriate subset of available actions. The same is true of blackboard systems (Hayes-Roth, 1985). Such strategies are also to be found buried in the details of some well-known BBAI architectures (for example, Blumberg, 1996).

Modularity by context must be vertical, because it is necessarily context and task specific. It also involves a significant amount of processing structure, encompassing from perception to action. For example, emotional states might be seen as contextual modules—the state of fear reduces the problem of behaviour arbitration to a choice between fight or flight, yet all of an animal's senses and all of its muscles are available to effect these actions.

Although this sort of modularity violates Fodor's notions both by the lack of parallelism and by the end-to-end nature of its control, these are similar to the violations already mentioned in describing behaviour-based AI and MASs. As such, the concept of temporal modularity might be interesting to evolutionary psychologists and other massive modularists.

Summary: Modularity in Brains

As usual, when looking at real evolved systems, the picture of modularity gets much messier in the brain. There is, nevertheless, strong evidence for modularity of some sort. Recall the definition of modularity I championed in the first section—there can be no question that various organs of the brain and various regions of organs have specialized, externally inaccessible representations and processes. However, task-based activation incorporates many disparate parts of the brain. Rather than dividing into horizontal *or* vertical modules, it seems that much of mental processing can be pictured as a crisscross of activations between horizontal *and* vertical modules. Even V1, long "known" to be the area of human neocortex associated with basic interpretations of retinal (visual) data, has now been implicated for learning Braille (a tactile but still spatial skill) in both blind and sighted patients (Kauffman et al., 2002). Interestingly, this also requires contextual modularity—sighted patients find learning Braille much easier if they are blindfolded.

Why should the brain be modular? I suspect that evolution has found it useful for the same reason as software engineers have (Parnas et al., 1985; Coad et al., 1997): to combat the explosive combinatorial complexity of searching for the right solution to hard problems. During evolution (for the species) and development (for the individual), preferred synaptic organizations are learned for recognizing regularities that are useful for controlling actions and achieving goals. These regularities may arise from the external environment, as communicated by the senses, or from neighboring neural systems—from the neuron's perspective, there is no difference. This view of the development of modularity in the individual is similar to that of Bates (1999) and to some extent to that of Karmiloff-Smith (1992), whose work emphasizes the developmental aspects of skill specialization. On the species level, it echoes Livesey's (1986) account of the evolution of brain organization.

Discussion

The bulk of this chapter has been about the current state of the art in modularity. In this discussion, I turn at least a bit more speculative. To begin with, I will describe my own work (alluded to earlier) in behaviour coordination. Then I will dive into a brief discussion of the extent to which I have failed to discuss many of the cognitive entities generally associated with mind.

Module Coordination and Structured Action Selection

Although my main personal motivation is to understand how brains and minds work, the bulk of my research to date has been into the management and design of modular AI systems. This is because I think that modular AI systems are a good platform for modelling and, thus, understanding natural intelligence (Bryson & Stein, 2001b; Bryson, 2001).

Here is a brief summary of my current conclusions about the engineering of modular AI systems:

- Semantic and task memory should be stored in specialized representations within modules. That is, specialized memory should be stored with the processes that exploit it. This is roughly consistent with the massive-modularist approach to psychology, and it is a slight elaboration on the position of BBAI, as described earlier.
- Ordering the behaviour of such modules is best done using a specialized, horizontal module for sequencing behaviour. This sequencing module is not a full planning system, but rather a system for running established reactive plans (see Bryson & Stein, 2001a, for further details). As such, it is similar to the interface between the basal forebrain in mammals and parts of the midbrain implicated with storing action sequences (see Carlson, 2000; Lonstein & Stern, 1997; Redgrave et al., 1999; and Mink, 1996).

These are the systems that I have the most experience of building myself, but I believe there are other horizontal modules that will be of general use that my systems are just not yet sophisticated enough to require. I suspect that it is useful to have coordination and smoothing of motor command conducted by a hindbrain-analogue module. There is already some evidence for this in the literature, for example, the motor command of the Ymir architecture (Thórisson, 1999) or the work of various groups in modelling the cerebellum. To date, some of the smoothest controls of complex robotics have come from monolithic "dynamical systems" based on mathematical control (for example, Atkeson et al., 1997), but I suspect that breaking this approach into modules will make it easier to scale it up to more complex tasks.

I also suspect that real agents need a hippocampus analogue. The hippocampus seems to be an organ with highly indexical, context-dependent relations, which rapidly learns associations. This organ has been implicated as necessary for episodic memory and for navigation. My current research is leading me to believe that these may follow from each other—that the organ may have originally been specialized to learning navigation but became adapted for task learning in general, and episodic memory may have become one of the features of this capability. Unfortunately, it may not be a good idea for artificial agents to be built to include a primate-like capacity for task learning, as such a system would necessarily slow the system's processing and reduce its reliability (Bryson & Hauser, 2002). But, it *is* clear that any humanlike mind would need this kind of capacity.

Deliberation

Episodic memory is occasionally called "declarative memory," because in humans, it is the kind of memory you can talk about (have conscious access to). This is a strange term in a sense, because there is evidence of other animals having memories of isolated experiences, but not of them declaring much of anything. Episodic memory is useful locally for keeping place within a task, for example, knowing which parts of a familiar maze you have explored already today (assuming the maze is rebaited everyday). It might be useful long term as a source of cases for case-based reasoning or as raw data to be compiled statistically into probability frameworks so that expectations can be used for planning or to disambiguate noisy sensory data.

On the other hand, the role of deliberation (or conscious attention to a task) still seems deeply mysterious to me. As already mentioned, the accessibility difference that determines explicit from implicit knowledge *is* a key indicator of modularity. But I see no systematic difference (other than qualia) between conscious and unconscious thought other than a marked increase in cortical activity (Dehaene et al., 2001, 1998).

Unlike some researchers in AI, I am not convinced that consciousness is isomorphic with having self-knowledge, although clearly having a good representation of oneself is useful to planning. Nor is it with having language. Language almost certainly fundamentally *alters* the nature of consciousness, both by allowing shorthand concept reference in what is clearly a limited-capacity system and by increasing coherence as a consequence of language's sequential temporal nature (Spelke, 2003; Bryson, 2002). I could easily construct an AI straw-being that might have either or both of these attributes but not seem particularly more alive or aware than any other AI system.

The consciousness-related data that are currently intriguing me most are a number of recent results from a variety of laboratories (Siemann & Delius, 1993; Greene et al., 2001; Bechara et al., 1995) showing the following:

- Humans can learn complex tasks without explicitly understanding them.
- Humans who gain an explicit understanding show no performance difference from those who do not.

I suspect that two things are true. First, I believe Dennett (2001) is absolutely right in suggesting that as we come to understand consciousness, we will realize that we have been covering several disparate functions with that one term, none of which are magic. Second, I believe that two of these functions will turn out to be focusing search for action selection and ordering behaviour in time.

Conclusions

This chapter has been an introduction to the idea of modularity as approached from three different disciplines: psychology, AI, and neuroscience. It has attempted to create a common framework for discourse between these fields, leveraging (but not necessarily supporting) the decomposition originated by Fodor (1983) between the notion of horizontal modules, which affect all of an agent's intelligence, and vertical modules, which are specialized skill areas showing disassociated abilities or deficits. Many topics have been touched on only lightly here, and hopefully the bibliography can fill in more details for anyone interested.

Modularity is a key aspect of mind—it explains or at least describes our inability to access all of our intelligence, which, in turn, justifies the hacks we use to control our own behaviour—for example, not having a whole box of cookies in the house or not having even one drink if we know it would lead to more drinks than it is safe to have before driving. Modularity also casts an interesting light on the fundamental mental problem of planning or action selection. If our minds are modular, then choosing the next action is not necessarily something done by visiting all possible alternatives but may instead be a matter of arbitrating between a number of alternative courses of action proposed by modules. These modules are then engaged in a similar problem, but over a more limited set of goals and possible courses of action.

In conclusion, I expect modular models of intelligence will continue to dominate both AI and the natural intelligences, though not necessarily with such clean and simplistic models as we began with. One of the current fundamental problems of AI is to enable the automatic learning of modular representations. This is difficult, because it requires a general-purpose representational substrate that will almost certainly be slow and inefficient. It should also be noticed that the brain is not a general-purpose representational substrate, but it incorporates an enormous amount of genetic bias that enables learning in animals. Similarly, as systems neuroscience comes to dominate the biological attempts to understand intelligence, there will be increased demands for models that somehow hide or abstract from the complexity of real, messy modules, yet allow for the interconnectivity that makes the whole thing work.

Acknowledgments

A short version of this chapter first appeared in a workshop coordinated by Alexander M. Meystel. Much of the content on psychology was gleaned from a seminar I was fortunate to attend led by Liz Spelke, Nancy Kanwisher, and Susan Carey. Thanks also to Aaron Sloman, David Glasspool, Dan Dennett, Push Singh, Dylan Evans, Peter Carruthers, Marc Hauser, Jon Flombaum, an anonymous reviewer, and, particularly, Will Lowe.

References

Albus, J. S. (1997). The NIST real-time control system (RCS): An approach to intelligent systems research. J*ournal of Experimental and Theoretical Artificial Intelligence*, *9*(2/3):147–156.

Anderson, J. R. (1993). *Rules of the mind.* Hillsdale, NJ: Lawrence Erlbaum Associates.

Arkin, R. C. (1998). *Behavior-based robotics.* Cambridge, MA: MIT Press.

Asaad, W. F., Rainer, G., & Miller, E. K. (2000). Task-specific neural activity in the primate prefrontal cortex. *Journal of Neurophysiology, 84*, 451–459.

Atkeson, C. G., Moore, A. W., & Schaal, S. (1997). Locally weighted learning for control. *Artificial Intelligence Review, 11*, 75–113.

Bates, E. (1999). Plasticity, localization and language development. In S. Broman & J. M. Fletcher (Eds.), *The changing nervous system: Neurobehavioral consequences of early brain disorders* (pp. 214–253). Oxford: Oxford University Press.

Bechara, A., Tranel, D., Damasio, H., Adolphs, R., Rockland, C., & Damasio, A. R. (1995). Double dissociation of conditioning and declarative knowledge relative to the amygdala and hippocampus in humans. *Science, 269*(5227), 1115–1118.

Blumberg, B. M. (1996). *Old tricks, new dogs: Ethology and interactive creatures.* PhD thesis. Cambridge, MA: MIT, Media Laboratory, Learning and Common Sense Section.

Bonasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., & Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence, 9*(2/3), 237–256.

Brooks, R. A. (1986). A robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation,* RA-2, 14–23.

Brooks, R. A. (1991a). Intelligence without reason. In *Proceedings of the 1991 International Joint Conference on Artificial Intelligence* (pp. 569–595). Sydney, Australia.

Brooks, R. A. (1991b). Intelligence without representation. *Artificial Intelligence, 47*, 139–159.

Brosnan, S. F., & de Waal, F. B. M. (2003). Monkeys reject unequal pay. *Nature, 425,* 297–299.

Bryson, J. J. (2000). Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence, 12*(2), 165–190.

Bryson, J. J. (2001). *Intelligence by design: Principles of modularity and coordination for engineering complex adaptive agents.* PhD thesis. Cambridge, MA: MIT, Department of EECS. AI Technical Report 2001-003.

Bryson, J. J. (2002). Language isn't quite that special. *Brain and Behavioral Sciences,* 25(6), 679–680. Commentary on Carruthers "The Cognitive Functions of Language," same volume.

Bryson, J. J. (2003). Where should complexity go? Cooperation in complex agents with minimal communication. In W. Truszkowski, C. Rouff, & M. Hinchey (Eds.), *Innovative concepts for agent-based systems* (pp. 298–313). Heidelberg: Springer.

Bryson, J. J., & Hauser, M. D. (2002). What monkeys see and don't do: Agent models of safe learning in primates. In M. Barley & H. W. Guesgen (Eds.), *AAAI Spring Symposium on Safe Learning Agents.*

Bryson, J. J., & Stein, L. A. (2001a). Modularity and design in reactive intelligence. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence* (pp. 1115–1120). San Francisco, CA: Morgan Kaufman.

Bryson, J. J., & Stein, L. A. (2001b). Modularity and specialized learning: Mapping between agent architectures and brain organization. In S. Wermter, J. Austin, & D. Willshaw (Eds.), *Emergent neural computational architectures based on neuroscience* (pp. 98–113). Heidelberg: Springer.

Carlson, N. R. (2000). *Physiology of behavior.* Boston, MA: Allyn & Bacon.

Carruthers, P. (2003). The cognitive functions of language. *Brain and Behavioral Sciences, 25*(6).

Chomsky, N. (1980). Rules and representations. *Brain and Behavioral Sciences, 3,* 1–61.

Coad, P., North, D., & Mayfield, M. (1997). *Object models: Strategies, patterns and applications* (2nd ed.). New York: Prentice Hall.

Coltheart, M. (1999). Modularity and cognition. *Trends in Cognitive Sciences, 3*(3), 115–120.

Connell, J. H. (1990). *Minimalist mobile robotics: A colony-style architecture for a mobile robot.* New York: Academic Press (also Cambridge, MA: MIT Press, TR-1151).

Cosmides, L., & Tooby, J. (1994). Origins of domain specificity: The evolution of functional organization. In L. A. Hirschfeld & S. A. Gelman (Eds.), *Mapping the mind: Domain specificity in cognition and culture.* London; New York: Cambridge University Press.

Dawkins, R. (1976). Hierarchical organisation: A candidate principle for ethology. In P. P. G. Bateson & R. A. Hinde (Eds.), *Growing points in ethology* (pp. 7–54). London; New York: Cambridge University Press.

Dehaene, S., Kerszberg, M., & Changeux, J. -P. (1998). A neuronal model of a global workspace in effortful cognitive tasks. *Proceedings of the National Academy of Science*, USA, 95, 14529–14534.

Dehaene, S., Naccache, L., Cohen, L., Le Bihan, D., Mangin, J. -F., Poline, J. -B., & Rivière, D. (2001). Cerebral mechanisms of word masking and unconscious repetition priming. *Nature Neuroscience, 4*(7), 678–680.

Dennett, D. C. (2001). Are we explaining consciousness yet? *Cognition, 79,* 221–237.

d'Inverno, M., Kinny, D., Luck, M., & Wooldridge, M. (1997). A formal specification of dMARS. In M. P. Singh, A. S. Rao, & M. J. Wooldridge (Eds.), *Proceedings of the Fourth International Workshop on Agent*

Theories, Architectures and Languages (pp. 155–176). Heidelberg: Springer.

Donnai, D., & Karmiloff-Smith, A. (2000). Williams syndrome: From genotype through to the cognitive phenotype. *American Journal of Medical Genetics, 97*(2), 164–171.

Downing, P. E., Liu, J., & Kanwisher, N. (2001). Testing cognitive models of visual attention with fmri and meg. *Neuropsychologia, 39*, 1329–1342.

Doyle, J. (1983). *A society of mind.* Technical Report 127, CMU Department of Computer Science.

Edmonds, B. (2002). A review of "reasoning about rational agents" by Michael Wooldridge. *Journal of Artificial Societies and Social Simulation, 5*(1). Retrieved from *http://jasss.soc.surrey.ac.uk/5/1/reviews/edmonds.html*

Elman, J. L., Bates, E. A., Johnson, M. H., Karmiloff-Smith, A., Parisi, D., & Plunkett, K. (1996). *Rethinking innateness. A Connectionist perspective on development.* Cambridge, MA: MIT Press.

Evans, D., & Zarate, O. (1999). *Introducing evolutionary psychology.* Cambridge: Icon Books Ltd.

Flombaum, J. I., Santos, L. R., & Hauser, M. D. (2002). Neuroecology and psychological modularity. *Trends in Cognitive Sciences, 6*(3), 106–108.

Fodor, J. A. (1983). *The modularity of mind.* Cambridge, MA: Bradford Books, MIT Press.

Freedman, D. J., Riesenhuber, M., Poggio, T., & Miller, E. K. (2001). Categorical representation of visual stimuli in the primate prefrontal cortex. *Science, 291,* 312–316.

Gall, F. J. (1825). *Sur l'origine des qualités morales et des facultés intellectuelles de l'homme, et sur les conditions de leur manifestation.* Paris: J. B. Baillière.

Gardner, M. (1970). Mathematical games: The fantastic combinations of John Conway's new solitaire game "Life." *Scientific American, 223*(4), 120–123.

Gat, E. (1991). *Reliable goal-directed reactive control of autonomous mobile robots.* PhD thesis, Virginia Polytechnic Institute and State University.

Gat, E. (1998). Three-layer architectures. In D. Kortenkamp, R. P. Bonasso, & R. Murphy (Eds.), *Artificial intelligence and mobile robots: Case studies of successful robot systems* (pp. 195–210). Cambridge, MA: MIT Press.

Greene, A. J., Spellman, B. A., Dusek, J. A., Eichenbaum, H. B., & Levy, W. B. (2001). Relational learning with and without awareness: Transitive inference using nonverbal stimuli in humans. *Memory & Cognition, 29*(6), 893–902.

Guzzoni, D., Cheyer, A., Julia, L., & Konolige, K. (1997). Many robots make short work. *AI Magazine, 18*(1), 55–64.

Hauser, M. D. (1999). Perseveration, inhibition and the prefrontal cortex: A new look. Current Opinion in Neurobiology, 9, 214–222.

Hayes-Roth, B. (1985). A blackboard architecture for control. *Artificial Intelligence, 26*(3), 251–321.

Hexmoor, H., Horswill, I., & Kortenkamp, D. (1997). Special issue: Software architectures for hardware agents. *Journal of Experimental and Theoretical Artificial Intelligence, 9*(2/3).

Hubel, D. H. (1988). *Eye, brain and vision.* New York: Freeman.

Hume, D. (1748). *Philisophical essays concerning human understanding.* London: Andrew Millar.

Karmiloff-Smith, A. (1992). *Beyond modularity: A developmental perspective on cognitive change.* Cambridge, MA: MIT Press.

Kauffman, T., Theoret, H., & Pascual-Leone, A. (2002). Braille character discrimination in blindfolded human subjects. *Neuroreport, 13*(5), 571–574.

Kobayashi, T., Nishijo, H., Fukuda, M., Bures, J., & Ono, T. (1997). Task-dependent representations in rat hippocampal place neurons. *Journal of Neurophysiology, 78*(2), 597–613.

Konolige, K., & Myers, K. (1998). The Saphira architecture for autonomous mobile robots. In D. Kortenkamp, R. P. Bonasso, & R. Murphy (Eds.), *Artificial intelligence and mobile robots: Case studies of successful robot systems* (chap. 9, pp. 211–242). Cambridge, MA: MIT Press.

Kortenkamp, D., Bonasso, R. P., & Murphy, R. (Eds.). (1998). *Artificial intelligence and mobile robots: Case studies of successful robot systems.* Cambridge, MA: MIT Press.

Livesey, P. J. (1986). *Learning and emotion: A biological synthesis* (Vol. 1 of Evolutionary Processes). Hillsdale, NJ: Lawrence Erlbaum Associates.

Lonstein, J. S., & Stern, J. M. (1997). Role of the midbrain periaqueductal gray in maternal nurturance and aggression: c-fos and electrolytic lesion studies in lactating rats. *Journal of Neuroscience, 17*(9), 3364–3378.

Malcolm, C., Smithers, T., & Hallam, J. (1989). An emerging paradigm in robot architecture. In *Proceedings of the International Conference on Intelligent Autonomous Systems (IAS)* (Vol. 2, pp. 545–564). Amsterdam; New York: Elsevier.

Mataric, M. J. (1990). *A distributed model for mobile robot environment-learning and navigation*. Technical Report 1228. Cambridge, MA: Massachusetts Institute of Technology, Artificial Intelligence Lab.

Mink, J. W. (1996). The basal ganglia: Focused selection and inhibition of competing motor programs. *Progress In Neurobiology, 50*(4), 381–425.

Minsky, M. (1985). *The society of mind.* New York: Simon & Schuster.

Newell, A. (1990). *Unified theories of cognition.* Cambridge, MA: Harvard University Press.

Parnas, D. L., Clements, P. C., & Weiss, D. M. (1985). The modular structure of complex systems. *IEEE Transactions on Software Engineering, SE-11*(3), 259–266.

Perrett, D. I., Hietanen, J. K., Oram, M. W., & Benson, P. J. (1992). Organisation and functions of cells responsive to faces in the temporal cortex. *Philosophical Transactions of the Royal Society of London, 335,* 25–30.

Redgrave, P., Prescott, T. J., & Gurney, K. (1999). The basal ganglia: A vertebrate solution to the selection problem? *Neuroscience, 89*, 1009–1023.

Sen, K., Theunissen, F. E., & Doupe, A. J. (2001). Feature analysis of natural sounds in the songbird auditory forebrain. *Journal of Neurophysiology, 86*(3), 1445–1458.

Sengers, P. (1999). *Anti-Boxology: Agent design in cultural context.* PhD thesis. Pittsburgh, PA: Carnegie-Mellon University, School of Computer Science.

Siemann, M., & Delius, J. D. (1993). Implicit deductive reasoning in humans. *Naturwissenshaften, 80,* 364–366.

Skaggs, W., & McNaughton, B. (1998). Spatial firing properties of hippocampal CA1 populations in an environment containing two visually identical regions. *Journal of Neuroscience, 18*(20), 8455–8466.

Sloman, A., & Logan, B. (1999). Building cognitively rich agents using the Sim_agent toolkit. *Communications of the Association of Computing Machinery, 42*(3), 71–77.

Spelke, E. S. (2003). What makes us smart? Core knowledge and natural language. In D. Gentner & S. Goldin-Meadow (Eds.), *Advances in the investigation of language and thought.* Cambridge, MA: MIT Press.

Tanaka, J. W., & Farah, M. J. (1993). Parts and wholes in face recognition. *Quarterly Journal of Experimental Psychology, 46A*(2), 225–245.

Thelen, E., & Smith, L. B. (1994). *A dynamical systems approach to development of cognition and action.* Cambridge, MA: MIT Press.

Thórisson, K. R. (1999). A mind model for multimodal communicative creatures and humanoids. *International Journal of Applied Artificial Intelligence, 13*(4/5), 519–538.

Weiß, G. (Ed.). (1999). *Multiagent systems.* Cambridge, MA: MIT Press.

Winston, P. (1975). Learning structural descriptions from examples. In P. Winston (Ed.), *The psychology of computer vision.* New York: McGraw-Hill.

Wooldridge, M. J., & Ciancarini, P. (2001). Agent-oriented software engineering: The state of the art. In P. Ciancarini & M. J. Wooldridge (Eds.), *First International Workshop on Agent-Oriented Software Engineering* (Vol. 1957 of LNCS, pp. 1–28). Heidelberg: Springer.

Endnote

[1] Most of this discussion is true of vertebrate brains in general, but I am most familiar with primate brains, so I restrict my claims.

Chapter 5

Memory and Emotion in the Cognitive Architecture

William F. Clocksin
Oxford Brookes University, UK

Abstract

This chapter explores issues in memory and affect in connection with possible architectures for artificial cognition. Because memory and emotion are evolutionarily and developmentally rooted in social interdependence, a new understanding of these issues is necessary for the principled design of true intelligent systems. We treat emotion not as an optional extra or as a brief episode of feeling, but as the underlying substrate enabling the formation of social relationships essential for the construction of cognition. We treat memory not as the storage and retrieval of immutable data, but as a continuous process contingent on experience and never fully fixed or immutable. Three converging areas of research are identified that hold some promise for further research: social constructionism, reconsolidation memory theory, and memory models based on the nonlinear dynamics of unstable periodic orbits. We argue that the combination of these ideas offers a potentially more substantive approach to understanding the cognitive architecture.

Introduction

This chapter is concerned with memory and affect in connection with possible architectures for artificial cognition. The work described in this chapter represents a departure from the traditional ways in which memory and emotion have been considered in artificial intelligence (AI) research and is informed primarily by two strands of thought emerging from social and developmental psychology. First, there has been increasing concern with personhood: with persons, agency, and action, rather than causes, behaviors, and objects (Shotter & Gergen, 1989). Second, there is an emphasis on the self as a social construct (Gergen, 1999), that persons are the result of interactions with significant others, and that the nature of these interactions is, in turn, shaped by the settings in which these interactions occur (Levine, 1992).

There is a tradition in cognitive science of teasing the cognitive architecture apart into components, so that their features and characteristics may be better understood. Examples of such components are memory, perception, problem solving, attention, and emotions. The reductionist method of decomposing a complicated problem into subproblems that can be studied individually has a long and distinguished history and has well served the physical sciences. However, it is important to remember that cognition takes place within communities and cultures and is found within a complex matrix of interactions and reciprocalities of needs and desires. Although the nominalist philosophy that has been the prevailing thoughtway for more than 300 years has privileged the individual mind as the centre of being and the locus of meaning-producing processes (Churchland, 1995), it is well to remember Weizenbaum's premise (1976) that intelligence is manifested only within a cultural and social milieu. As Aubé concludes elsewhere in this book, it is incoherent to try to design emotional systems for artifacts that do not belong to communities.

It is possible that an understanding of cognition rich enough to provide a computational basis for artificial intelligence will need to depend on understanding the human person, which in turn, depends on understanding the propinquity of interaction between persons. The reductionist and nominalist projects do not offer adequate means of obtaining a conceptual grasp of the complex ecosystem of interactions within communities of persons. As Fraser Watts (2000) said, the human emotions illustrate very well the way in which the biological and social aspect of personhood are intertwined. Emotions have both biological and social aspects, and any attempt to explain one aspect without the other leads to an impoverished account.

The main objective of this chapter is to show how ideas from three different research areas might be caused to converge in order to form a new way to think about cognition. The motivation for forcing this convergence is the need to look beyond the Jamesian tradition of understanding emotion and memory as relatively brief episodes of an individual's experience. Instead, memory and emotion are seen here as partners that coconstruct an interactive involvement between persons. This involvement is extended in time and has a character that is provisional, circular, changing, and never finished. This involvement can be pictured as a rope made of three different strands of research. The first strand is social constructionism (Gergen, 1999). Although most work in the area has taken place only within the past 20 years, its emphasis on the self and meaning as achievements emerging from social interactions can be traced back to G. H. Mead (1934). Social constructionism can provide a way to break out of the habit of thinking of cognition as a solely individual affair governed by the physical processes within an individual's brain, and it emphasizes the circular relationship between interpretation and understanding. It is important to understand this circular relationship as being supported and enabled by brain functions, and this motivates the inclusion of the second and third strands. The second strand is neurobiology, from which a new memory model called reconsolidation has been proposed (Nader, 2003). Reconsolidation is the hypothesis that a memory can be returned to a mutable, sensitive state in which it can be modified, strengthened, changed, or even erased. New studies support a reconsolidation theory, in which the traditional distinction between long-term memory and short-term memory must be reconsidered. The third strand is a branch of nonlinear dynamics referred to as chaotic computation, from which has recently emerged a mathematical model describing how memories may be represented as unstable orbits within a phase space, thus providing theoretical underpinnings for a mutable memory model (Crook & olde Scheper, 2002). In this approach, memories are not stored as distributed patterns of weights in a network of neural units, as in well-known connectionist models. Instead, memory states emerge from the dynamics of the network. The appeal of this model is that the self-organized representations that emerge are a consequence of the resolution of internal and external dynamic effects over a period of time. This seems to be consistent with a reconsolidation memory model. It also seems to be consistent with the reciprocal meaning-generating interaction emphasized by the social constructionist approach as well as with a mechanism to explain its foundations in neural function.

Background

Current thinking on cognitive architecture can be identified with two terms: situated cognition and the dynamical hypothesis. This section briefly addresses these areas, and then turns to the other topics—social construction, emotion, memory—that provide the conceptual background of this chapter.

Situated Cognition

Situated cognition (Clancy, 1997) is a research approach that relates the activities of the cognitive agent with the environment within which the agent's activities take place. The word "situated" is not meant to imply that cognition is fixed to localized situations. Instead, situated cognition emphasizes that there is an inescapable environmental context for cognitive activity, and that cognition takes place within a totality of activity, including social activity. Situated cognition can be contrasted to traditional information-processing (TIP) views of cognition (for example, Newell and Simon, 1972), which focus on symbolic representations of mind and the mechanisms that operate on symbols. Assuming that cognitive processing takes place within the brains of individuals, the TIP view seeks to understand internal mental processes, and it models the individual as an input/output function. By contrast, situated cognition focuses on the structures of the world and how they constrain behavior. Many of the influences on situated cognition can be traced back to Neisser (1976) and the ecological psychology of Gibson (1979, 1966).

Situated cognition can be approached from two perspectives: one with an internal focus on mechanisms of the mind, and one with an external focus on the social community. The first perspective, taken by many cognitive science and AI researchers, is committed to understanding perception-action systems, and is expressed in AI research as situated robotics (Brooks, 1991). However, Clancy (1995) argued that situated robotics is different from situated cognition, because situated robotics is too much like TIP in focussing on the contextualized perception-action problem at the expense of understanding social activity as an integral part of cognition.

The other perspective (Lave, 1988) sees situated cognition from an almost entirely social or cultural standpoint. Knowing, learning, and cognition are social constructs, expressed through the actions of people interacting in communities. Cognition is constructed through these actions. This approach avoids talking of mental mechanisms and focuses instead on the context of actions and behaviors within the social group. This perspective on situated cognition has had significant influence on new approaches to human learning (Brown et al., 1989; Lave & Wenger, 1990).

Both perspectives would agree that the situatedness of cognition is a consequence of embodiment. The embodied nature of cognition (Wilson, 2002) suggests that cognition is not a mental machine working on abstract problems, but it is an activity connected with a body that requires cognition to make it function.

Finally, it is fair to say that the various perspectives on situated cognition are like TIP in not providing a focus on the role of emotions in behavior. To the extent that emotions are considered in situated cognition, there seems to be an underlying assumption of a *natural kind* for emotion (Charland, 1995), that is, that emotion exists as a distinct natural domain governed by its own representations and mechanisms. Griffiths (2002) argued against this position, suggesting there is no such thing as a typical emotion and that emotions result from different psychological and neurological mechanisms. The approach we take in this chapter is to consider emotions of primary importance, because they enable the human relationships within which cognition is situated. Although Charland defended a similar hypothesis that emotions may be more fundamental in the organization of behavior than cognition, this chapter concurs with Griffith's scepticism about identifying emotions with a unique form of representation and computation.

The Dynamical Hypothesis

The dynamical hypothesis is a term coined by van Gelder (1998), who described it by the slogan "cognitive agents are dynamical systems." Cognition is considered a dynamical phenomenon and is best understood in dynamical terms. A dynamical system is a set of variables changing continually and interdependently over time in accordance with dynamical laws described by a set of equations. Dynamical models see representations as "transient, context-dependent stabilities in the midst of change, rather than as static, context-free, permanent units" (van Gelder, 1999). The main contribution of the dynamical hypothesis is to advocate the study of cognitive processes using the tools of dynamical systems theory. A weakness of the dynamical hypothesis is that it does not answer the key questions of "what types of dynamical systems are related to cognition, what types are not, and why" (French & Thomas, 2001). In this sense, the dynamical hypothesis is like a skeleton of theoretical ideas and hopes waiting for the flesh of empirical connections.

There are a number of points of contact between the dynamical hypothesis and many of the chapters in this book. This chapter points to a way of integrating memory and emotion in a way that has all the characteristics of a dynamical approach. The chaotic dynamics technique discussed below is one example, though the dynamical hypothesis does not depend upon chaotic dynamics in particular.

Social Construction

Our perspective on memory and emotion makes use of a way of thinking about persons known as *social constructionism*. It is useful at the outset to distinguish between construc*tivism* and construc*tionism*, though it should be noted that not all accounts observe this neat distinction of terminology. The main idea of constructivism (Piaget, 1990) is not dissimilar to that of TIP: understanding is created through the construction of a variety of mental schemata and procedures for analysis and synthesis of schemata. This is the main assumption behind cognitive psychology, and it is safe to say this is closely related to the prime, if unexpressed, assumptions of most traditional AI research. Papert (1991), who was a student of Piaget, used constructionism to mean the idea that constructing artifacts is a good way to learn, but this is not the meaning used in this chapter.

Social constructionism—inaccurately called social constructivism or even social constructionalism in some secondary sources—describes the work of Gergen (1994, 1999), Potter (1996), Shotter (1993), and Harré (1992). Social constructionism is concerned with the processes by which human abilities, experiences, common sense, and scientific knowledge are both produced in, and reproduce, human communities (Shotter & Gergen, 1994). The idea is that the processes that construct meaning are to be found in relationships, often discursive, between persons. Social constructionism also relates to John Searle's study of what he calls "institutional facts" (1995). A diversity of views on the social construction of emotions is surveyed by Harré (1986).

According to familiar AI models, the intelligent agent operates a perception–processing–action loop in order to solve problems that are presented to it. Meaning is constructed in the mind as a result of internal symbol-processing capability. By contrast, the constructionist approach privileges social interaction as the meaning-constructing process. Here, the mind is for social development, engagement with persons, and the institution and appropriation of personhood and identity. According to Gergen (1991):

> *In this way, meaning is born of interdependence. And because there is no self outside a system of meaning, it may be said that relation[ship]s precede and are more fundamental than self. Without relationship there is no language with which to conceptualize the emotions, thoughts, or intentions of the self. (p. 157, his brackets)*

Emotion

A new field is forming in computer science, named affective computing and defined as "computing that relates to, arises from, or deliberately influences emotions" (Picard, 1997). The premises of affective computing still have not been universally accepted within the AI research community. To understand why this is, we may identify three attitudes to the understanding of emotions in the context of AI research. These attitudes developed in roughly chronological order, though there are always individual exceptions to the chronology. More importantly, these attitudes are mindsets that describe and constrain the ways that researchers consider the role of emotion in intelligent systems. The first attitude, represented by what Haugeland (1986) has called "Good Old Fashioned AI" (GOFAI), holds the traditional dualist distinction between reasoning and the emotions. Because intelligence is seen as based on logical reasoning or abstract problem solving, emotions are undesirable and can only lead to distraction or error. This attitude is also expressed by popular science fiction, in which supreme reasoning beings are free of emotions: The android Mr. Data (in the television series *Star Trek: The Next Generation*) has been designed without emotions, and the half-human half-Vulcan Mr. Spock (of the original *Star Trek* television series) has been trained from birth to suppress emotions in the service of the logical thinking prized by the civilization on his home planet Vulcan. The second attitude considers emotion either as a tool or as a generated side effect of cognitive processes. This is a functionalist approach, in which emotions are seen as useful capabilities for making machines easier to use (Picard, 1997) or for making cognition more efficient, for example, by enabling the intelligent system to react more quickly to threats (Sloman, 2001). The second attitude is also expressed in *Star Trek: The Next Generation*, where Mr. Data has an emotions chip he can insert into his positronic brain when it is necessary to experience emotions to better perform some task. Most of the time, he prefers not to use the chip, regarding emotions as useful in some situations but not necessary. An early paper that can be identified with the second attitude was published during the height of the GOFAI era (Sloman & Croucher, 1981), thus representing one of the exceptions to the chronological ordering. The second attitude has been influenced mainly by cognitive psychology studies of emotion, for example, by Ortony et al. (1990), and is responsible for improving the public visibility of the issues connected with emotions and reasoning (Damasio, 1994; Jáuregui, 1995). The main contribution of the second attitude is the way it characterizes emotions in the context of cognitive function, instead of simply describing the emotions and emotion-related processes local to specific times and groups. In the third attitude, emotions are seen as the foundation on which all cognition and behavior are built. It starts from the premise that intelligence manifests itself only relative to specific social and cultural contexts (Weizenbaum, 1976) and is informed by recent progress in social psychology. Emotions constitute an ever-present substrate or foundation to everyday intelligent behaviors. Because cognition and emotion are seen to be constructed within a social context (Clocksin, 2003; Harré & Parrott, 1996), they are intrinsically linked. Seeing emotions in terms of a functional purpose (the second attitude) may be useful anthropology, but this is not the point emphasized by the third attitude. Instead, intelligence and meaning are constructed within social relationships, and emotions are the uniquely human capability that enables complex social relationships to happen and enables relationships to be expressed though conversation and narrative. This can be studied in various ways. For example, because spontaneous speech is rarely free of emotion, recent work on the emotional states expressed through speech (Cowie & Cornelius, 2003) underscores the importance of emotion as an enabler of social interaction. Like the first and second attitude, the third attitude is also expressed in popular science fiction. In the film *The Bicentennial Man*, a robot develops over a 200-year time span to become fully human as a result of relationships with successive generations of a family. In the film *AI: Artificial Intelligence*, a *true* intelligent android is distinguished from other androids by his capacity to love and be loved.

Memory

There are two general strands of research about the social construction of memory. Both strands have, in common, the premise that human memory can only function within a collective context. The first strand is traced back to the work of Maurice Halbwachs (1877–1945) and is primarily concerned with collective memory over a historical timescale: how we use our experience of the present to reconstruct our historical past, and how this reconstructed past is used to guide our behavior in the present (Halbwachs, 1992). Collective memory is selective: different groups of people have different collective memories, which in turn give rise to different modes of behavior. Halbwachs shows, for example, how wealthy old families in France have a memory of the past that diverges sharply from that of the *nouveaux riches*, and how working class constructions of reality differ from those of their middle-class counterparts. The second strand of research is traced back to Lev Vygotsky (1896–1934), who emphasized the social dimensions of knowledge construction: relations between people are fundamental to all higher mental functions (ET 1978). This strand tends to focus on the timescale of the individual life—how our memories are formed and changed through our relationships with others.

From an experiential perspective, memory is often associated with episodes in which strong feelings of recall briefly dominate a person's awareness. From a functional perspective, memory is seen as a capability for storage and retrieval of data. However, here we consider memory in a broad sense that pervades experience, communication, and cognition. Furthermore, it is difficult to justify separating memory from emotion. We have feelings about what we remember, and these feelings strongly influence the ways we plan and learn. Feelings affect what we remember, and feelings affect how we remember.

Memory consists of much more than what can be understood from local feelings of recall. The experience of memory forms parts of ongoing interactions that are emotionally charged and are embedded in a broad social context. Significant memories are, to a large extent, social phenomena that take place in specific cultural settings. A memory is never a single state: it is an experience.

The relationship between memory and emotion is beginning to receive much attention at all levels, from behavior to neurophysiology (Rolls, 2003; Reisberg & Hartel, 2004; McGaugh, 2003). Emotional events are typically distinctive events, and distinctive events are easily retrieved from memory. However, strong emotion appears to disrupt memory processing, leading to a narrowing of processing and a loss of memory for peripheral details.

Following Shotter (1990), we see memory as constructed within the context of social relationships. Memories cannot be decoupled from the social experience of remembering, for outside the social context, they have no meaning. Furthermore, the conventional view of memory as storage-and-recall tends to neglect the importance of forgetting as part of the process of remembering. As Platinga (1992) put it, "our memories are not inert, but undergo a process of editing, whereby they are regularized, rendered more retainable and reshaped with an eye to subsequent circumstances and events" (p. 45). The more recently explored phenomena of *false memory* (Loftus, 1997) and the *misinformation effect* (Loftus & Hoffman, 1989) can thereby be understood from a social construction perspective. In the next section, we will see how recent neurobiology and theoretical models suggest that the storage-and-recall model of memory should be reconsidered in favor of a more cyclic model that is committed to the idea of memory as the continual remembering of our memories of our memories (sic).

Convergence

One theme of this chapter might be described as circularity. We can identify a circularity of process at all levels of behavior. The *hermeneutical circle*, a concept that can be traced to Martin Heidegger (1996), is one example: Understanding and interpretation influence each other in a circular way. To understand a text, we need to interpret it. But, a certain understanding may lead us to consider a fresh interpretation. When the text is read in the light of the new interpretation, it may change our understanding of it. This may call for a revised interpretation, and so on. Such circles may occur several times during the reading and re-reading of a text. This example can be extended to having a conversation. Conversations normally progress not as an orderly sequence of question and answer, but as a tangle of starts and re-starts, as both interlocutors negotiate respective hermeneutical circles to arrive at a coconstructed meaning. The concept can be extended to social interaction. The behavior of the intelligent system depends on the challenges that arise from being embodied and being expected to participate in social interactions. There is a circular effect as new capabilities bring about more social experience, which in turn, modifies the capabilities for further social experience.

We consider memory to be a circular process of remembering and forgetting, as memories are continually edited and revised as they are used. This circular model of memory seems to be opposed by the standard theories of short-term memory and long-term memory, in which memory is information that is stored and later retrieved. However, more support for a circular model of memory comes from three converging directions: (a) at the psychological level, the social construction of memory, in which memory is seen as a dynamic social practice (Middleton & Edwards, 1990; Neisser & Fivush, 1994); (b) at the physiological level, memory as a continual process of reconsolidation (Nader, 2003); and (c) at the neural network level, where new models based on chaotic dynamics represent memory as unstable periodic orbits that can be altered by small fluctuations (Crook et al., 2003; Crook & olde Scheper, 2002). These three independent approaches seem to converge to a common model for memory that is different than traditionally held models of memory.

The purpose of a circular model is to implement a capacity for continual coevolution (of an understanding, of a conversation, of behavior, of remembering), or ontogenesis (Oyama, 1985). This is characteristic of a sustaining system, and such a sustaining system is, in turn, characteristic of a functioning mind and any living system. In contrast, AI research has primarily focused on *one-shot* models, in which intelligent activity reduces to finding the answer to a problem within a logically consistent system. I previously surveyed some criticisms of this approach (Clocksin, 1998, 1995). One-shot models prevail in folk psychology and have a long history. The one-shot model of learning, that is, learning as simply remembering for later recall, can be traced to Plato (ET, 2003), with further development by Augustine (ET, 2001). However, ontogenesis of the type proposed here may proceed for a system in which problems and answers are not neatly defined, and consistency in a logical sense is never achievable. Thus, there needs to be an emphasis on the provisionality of thought and behavior, another hallmark of a social constructionist perspective.

Reconsolidation

The standard theory about human memory is that new memories are initially in a dynamic *labile* form for a short time (short-term memory), after which a memory trace is *fixed* or *consolidated* into the physical structure of the brain (long-term memory). During the past 40 years, much research has described the processes that contribute to the transformation of a memory trace from being mutable to becoming fixed. The short-term/long-term memory model has been taken as the unquestionable foundation of memory models in cognitive science and artificial intelligence. However, a number of recent studies (Nader, 2003) have challenged the view that memories are consolidated into a permanent state. Experiments using the technique of electrically induced amnesia have been interpreted to suggest that the reactivation of a consolidated memory returns it to a labile state, which initiates another time-dependent memory process similar to that seen after new learning (Lewis, 1979). This phenomenon is now referred to as reconsolidation, and findings have been replicated and extended in a variety of species, from slugs to humans, and across experimental setups. The continuous cyclic process of reconsolidation has been proposed not only at the cellular level, in which cells both in the hippocampus and neocortex apparently cycle between active states and inactive states, but also at a systems level, where a reconsolidation cycle seems to occur between the hippocampus and neocortex.

That memory is a fundamentally dynamic process was demonstrated by Bartlett (1932). As Nader put it, memories are not snapshots of events that are passively read out, but they are constructive in nature and always changing. How strange, therefore, that the standard memory model emphasizes the fixation of memory, and that error-free storage-and-retrieval is the gold standard of generations of psychological experiment. Given the one-shot nature of the storage-and-retrieval model, perhaps it has taken us as far as it can. By contrast, the reconsolidation model is a mechanism that makes explicit the dynamic character of memory and neural systems. It is not easy to correlate an organism's behavior and neural activity, but it could be said that if the dynamic, circular character of the social constructionist perspective on behavior needed to point to a neural model for support, then the reconsolidation model seems to have the right properties. The relationship between memory and emotion was introduced earlier, but now we relate memory and emotion to the theme of circularity. The extended process of reconsolidation suggests that the consolidation of memory may be a slow process. McGaugh (2003) explained how the slow consolidation of memory allows physiological processes activated by experiences to regulate the strength of the memory of the experiences. Emotionally arousing experiences induce the release of stress hormones that act on the brain to influence the consolidation of our memories of recent experience. This process takes time and may involve more than one cycle of reconsolidation. At the neurophysiological level (Rolls, 2003), emotion may influence the storage and recall of memories. Amygdala back-projections to the neocortex (Rolls & Treves, 1998) could perform this for emotion in an analogous way to the memory cycle between the hippocampus and neocortex. Thus, both memory and emotion seem to be involved in different but interrelated cycles of brain activity.

Chaotic Neural Models

If, as we suspect, the circular, provisional nature of behavior is a pervasive design characteristic in natural systems, it is desirable to model this in mathematical form. One possibility that seems to offer suitable mathematical tools is the branch of nonlinear systems known as chaos theory (Ott, 1993; Kaplan & Glass, 1995). The word "chaos" is commonly associated with disorder and has negative connotations. However, here we use its precise mathematical meaning: a chaotic system is one that has states that are deterministic but not predictable. Even simple nonlinear systems, such as oscillators, exhibit chaotic behavior. The behavior of a chaotic system may be represented by a trajectory through a phase space in which one or more attractors are embedded. The trajectory may take the form of an unstable periodic orbit (UPO) around an attractor. A given UPO can represent the memory state of a system. Chaotic systems offer several advantages to the engineer. One is that chaotic systems can be significantly easier to control than other linear or nonlinear systems, requiring only small, appropriately timed perturbations to constrain them to within specific UPOs. Another advantage is that chaotic attractors contain an infinite number of UPOs (though there is no guarantee they can all be stabilized), resulting in very high memory capacity compared to, say, the same number of weighted connections in a standard connectionist neural model. If individual UPOs can be made to represent specific internal states of a system, then a chaotic attractor can form part of a nearly infinite state machine. The key to making this work is to make small time-dependent perturbations in the orbits in a controllable manner, which may kick an orbit into another distinct orbit, or stabilize an orbit during a period of time.

Recent work (Crook et al., 2003; Crook & olde Scheper, 2002) has developed a working chaotic neural network that is able to select internal dynamic states in response to external inputs in a self-organized manner. Using this method, memories are not stored as distributed patterns of weights, as are commonly used with artificial neural networks. Instead, memory states emerge from the dynamics of the network. Conventional one-shot or incremental approaches to network adaptation cannot be applied to the chaotic model. Instead, network parameters are modified to perturb the orbits in order to support the dynamics from which memory states emerge. The work of Crook and olde Scheper demonstrates the ability of a chaotic system to respond to periodic input by stabilizing a UPO. The system can stabilize into different orbits depending on the presented input. The dynamic change of memory states involves the perturbation of orbits. One of the appealing aspects of this model is that the representations that emerge are a consequence of the resolution of internal and external dynamic states.

One cannot overlook the possibility of a link between the reconsolidation of a memory in the sense described by Nader and the stabilization of a UPO within a suitably configured chaotic system. It remains to be seen whether UPOs offer nothing more than a metaphor for the cyclic progress of reconsolidation, or whether the mechanisms underlying reconsolidation can be modelled as a chaotic system. At this stage, it is an intriguing prospect that chaotic dynamics in the manner explored by Crook and olde Scheper may offer a theoretical foundation to explain reciprocal, cyclic behavior over a wide range of levels of analysis, from the reconsolidation of memory and emotion to the conduct of a conversation within a socially constructed world. There has been no shortage of attempts to model a wide variety of biological systems using chaotic dynamics, and it is tempting to conclude that chaos is one of those *theories of everything* that turns out to explain nothing. However, the unique contribution of Crook and olde Scheper is to provide a constructive technique for describing the circular, provisional nature of state changes in terms of controlling UPOs by both internal and external inputs. This has clear implications for the design of systems that exhibit the provisional, circular behavior explored in this chapter.

Conclusion

Because memory and emotion are evolutionarily and developmentally rooted in social interdependence, a thorough exploration of these issues is necessary for the principled design of the cognitive architecture of true intelligent systems. In particular, we focused on two aspects of the cognitive architecture: emotion and memory. We have seen emotion not as an optional extra, or as discrete states or episodes of feelings, but as the underlying substrate enabling the formation of social relationships essential for the construction of cognition. We have seen memory not as the storage and retrieval of immutable data, but as a continuous process contingent on experience and never fully fixed or immutable. Memory and emotion seem to be inextricably connected at all levels, from the neurophysical to the social.

We identified three attitudes to the understanding of emotions in the context of AI research. We feel that AI research will be able to make more progress when it moves away from the ideas that emotions are either unnecessary or mere functional conveniences to the idea that the meaning-producing process is based on a propinquity of interaction between persons that is enabled by an affective foundation in each person's cognitive architecture.

Finally, it is hoped that the three intertwined strands of research discussed here—social constructionism, reconsolidation memory theory, and UPO memory models—may contribute to the growth of a new perspective on artificial intelligence surveyed by Clark (2001), and also may offer a specific focus on the details sought by researchers interested in dynamic approaches to cognition (van Gelder, 1999).

References

Aubé, M. (2004). Beyond needs: Emotions and the commitments requirement. In D. Davis (Ed.), *Visions of mind: Architectures for cognition and affect*. Hershey, PA: Idea Group.

Augustine. (2001). Confessions of St Augustine (trans. R. Warner). New York: Signet Classics.

Bartlett, F. (1932). *Remembering.* London; New York: Cambridge University Press.

Brooks, R. (1991). Intelligence without representation. *Artificial Intelligence, 47*(1), 139–159.

Brown, J., Collins, A., & Duguid, S. (1989). Situated cognition and the culture of learning. *Educational Researcher, 18*(1), 32–42.

Charland, L. (1995). Emotion as a natural kind: Towards a computation foundation for emotion theory. *Philosophical Psychology, 8*, 59–85.

Churchland, P. (1995). *The engine of reason, the seat of the soul.* Cambridge, MA: MIT Press.

Clancy, W. (1995). A boy scout, Toto, and a bird: How situated cognition is different from situated robotics. In L. Steels & R. Brooks (Eds.), *The artificial life route to artificial intelligence: Building situated embodied agents* (pp. 227–236). Hillsdale, NJ: Lawrence Erlbaum Associates.

Clancy, W. (1997). *Situated cognition: On human knowledge and computer representations.* London; New York: Cambridge University Press.

Clark, A. (2001). *Being there: Putting brain, body, and world together again.* Cambridge, MA: MIT Press.

Clocksin, W. (1995). Knowledge representation and myth. In J. Cornwell (Ed.), *Nature's imagination* (pp. 190–199). Oxford: Oxford University Press.

Clocksin, W. (1998). Artificial intelligence and human identity. In J. Cornwell (Ed.), *Consciousness and human identity* (pp. 101–121). Oxford: Oxford University Press.

Clocksin, W. (2003). Artificial intelligence and the future. *Philosophical Transactions of the Royal Society A, 361*, 1721–1748.

Cowie, R., & Cornelius, R. (2003). Describing the emotional states that are expressed in speech. *Speech Communication, 40*(1), 5–32.

Crook, N., & olde Scheper, T. (2002). Adaptation based on memory dynamics in a chaotic neural network. *Cybernetics and Systems, 33*(4), 341–378.

Crook, N., olde Scheper, T., & Pathirana, V. (2003). Self-organised dynamic recognition states for chaotic neural networks. *Information Sciences, 150*, 59–75.

Damasio, A. (1994). *Descartes' error: Emotion, reason and the human brain.* New York: Grosset/Putman.

French, R., & Thomas, E. (2001). The dynamical hypothesis in cognitive science: A review essay of Mind as Motion. *Minds and Machines, 11*(1), 101–111.

Gergen, K. (1991). *The saturated self: Dilemmas of identity in contemporary life.* New York: Basic Books.

Gergen, K. (1994). *Realities and relationships: Soundings in social construction.* Cambridge, MA: Harvard University Press.

Gergen, K. (1999). *An invitation to social construction.* London: Sage.

Gibson, J. (1966). *The senses considered as perceptual systems.* Boston, MA: Houghton Mifflin.

Gibson, J. (1979). *The ecological approach to visual perception.* Boston, MA: Houghton Mifflin.

Griffiths, P. (2002). Is emotion a natural kind? In R. Solomon (Ed.), *Philosophers on emotion.* Oxford: Oxford University Press.

Halbwachs, M. (1992). *On collective memory* (Ed., L. A. Coser). Chicago, IL: University of Chicago Press.

Harré, R. (1986). *The social construction of emotions*. Oxford: Blackwell.

Harré, R. (1992). *Social being: A theory for social psychology*. Oxford: Blackwell.

Harré, R., & Parrott, W. (1996). *The emotions: Social, cultural and physical dimensions.* London: Sage.

Haugeland, J. (1986). *Artificial intelligence: The very idea.* Cambridge, MA: MIT Press.

Heidegger, M. (1996). *Being and time* (trans. J. Stambaugh). Albany, NY: State University of New York Press.

Jáuregui, J. (1995). *The emotional computer.* Oxford: Blackwell.

Kaplan, D., & Glass, L. (1995). *Understanding nonlinear dynamics.* Heidelberg: Springer-Verlag.

Lave, J. (1988). *Cognition in practice.* London; New York: Cambridge University Press.

Lave, J., & Wenger, E. (1990). *Situated learning: Legitimate peripheral participation.* London; New York: Cambridge University Press.

Levine, G. (1992). *Constructions of the self.* New Brunswick, NJ: Rutgers University Press.

Lewis, D. (1979). Psychobiology of active and inactive memory. *Psychological Bulletin, 86,* 1054–1083.

Loftus, E. (1997). Creating false memories. *Scientific American, 227*(3), 70–75.

Loftus, E., & Hoffman, H. (1989). Misinformation and memory, the creation of new memories. *Journal of Experimental Psychology: General, 118*(1), 100–104.

McGaugh, J. (2003). *Memory and emotion.* New York: Columbia University Press.

Mead, G. (1934). *Mind, self and society.* Chicago, IL: University of Chicago Press.

Middleton, D., & Edwards, D. (1990). *Collective remembering.* London: Sage.

Nader, K. (2003). Memory traces unbound. *Trends in Neurosciences, 26*(2), 65–72.

Neisser, U. (1976). *Cognition and reality.* San Francisco: W.H. Freeman.

Neisser, U., & Fivush, R. (1994). *The remembering self: Construction and accuracy in the self-narrative.* London; New York: Cambridge University Press.

Newell, A., & Simon, H. (1972). *Human problem solving.* New York: Prentice Hall.

Ortony, A., Clore, G., & Collins, A. (1990). *The cognitive structure of emotions.* London; New York: Cambridge University Press.

Ott, E. (1993). *Chaos in dynamical systems.* London; New York: Cambridge University Press.

Oyama, S. (1985). *The ontogeny of information.* London; New York: Cambridge University Press.

Papert, S., & Harel, I. (1991). *Constructionism*. Norwood, NJ: Ablex.

Piaget, J. (1990). *The child's conception of the world.* New York: Littlefield Adams.

Picard, R. (1997). *Affective computing.* Cambridge, MA: MIT Press.

Platinga, T. (1992). *How memory shapes narratives.* Lampeter: Mellen.

Plato. (2003). *The last days of Socrates* (trans. H. Tredennick). New York: Penguin Books.

Potter, J. (1996). *Representing reality: Discourse, rhetoric and social construction.* London: Sage.

Reisberg, D., & Hartel, P. (Eds.). (2004). *Memory and emotion.* Oxford: Oxford University Press.

Rolls, E. (2003). Vision, emotion and memory: From neurophysiology to computation. In T. Ono, G. Matsumoto, R. R. Llinas, A. Berthoz, R. Norgren, H. Nishijo, & R. Tamura (Eds.), *Cognition and emotion in the brain.* Amsterdam; New York: Elsevier.

Rolls, E., & Treves, A. (1998). *Neural networks and brain function*. Oxford: Oxford University Press.

Searle, J. (1995). *The construction of social reality.* London: Penguin.

Shotter, J. (1990). The social construction of remembering and forgetting. In D. Middleton & D. Edwards (Eds.), *Collective remembering* (pp. 120–138). London: Sage.

Shotter, J. (1993). *Conversational realities: Constructing life through language.* London: Sage.

Shotter, J., & Gergen, K. (1989). *Texts of identity.* London: Sage.

Shotter, J., & Gergen, K. (1994). Series preface. In T. Sarbin & J. Kitsuse (Eds.), *Constructing the social* (p. i). London: Sage.

Sloman, A. (2001). Beyond shallow models of emotion. *Cognitive Processing, 2*(1), 177–198.

Sloman, A., & Croucher, M. (1981). Why robots will have emotions. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence* (pp. 197–202). Vancouver B.C., Canada.

van Gelder, T. (1998). The dynamical hypothesis in cognitive science. *Behavioral and Brain Sciences, 21*(5), 615–665.

van Gelder, T. (1999). Dynamic approaches to cognition. In R. Wilson & F. Keil (Eds.), *The MIT encyclopedia of cognitive sciences* (pp. 244–246). Cambridge, MA: MIT Press.

Vygotsky, L. (1978). *Mind in society: The development of higher psychological processes* (Eds., A. Blunden & N. Schmolze). Cambridge, MA: Harvard University Press.

Watts, F. (2000). The multifaceted nature of human personhood: Psychological and theological perspectives. In N. Gregersen, W. Drees, & U. Görman (Eds.), *The human person in science and theology* (pp. 41–63). Edinburgh: T and T Clark.

Weizenbaum, J. (1976). *Computer power and human reason.* Cambridge, MA: MIT Press.

Wilson, M. (2002). Six views of embodied cognition. *Psychonomic Bulletin and Review, 9*(4), 625–636.

Endnote

This chapter is dedicated to the memory of friend and colleague Dr. Marie Rosalie Dalziel (1939–2003), technology visionary and entrepreneur.

Chapter 6

Implementing Free Will

Bruce Edmonds
Manchester Metropolitan University, UK

Abstract

Free will is described in terms of the useful properties that it could confer, explaining why it might have been selected for over the course of evolution. These properties are exterior unpredictability, interior rationality, and social accountability. A process is described that might bring it about when deployed in a suitable social context. It is suggested that this process could be of an evolutionary nature—that free will might "evolve" in the brain during development. This mental evolution effectively separates the internal and external contexts, while retaining the coherency between individual's public accounts of their actions. This is supported by the properties of evolutionary algorithms and possesses the three desired properties. Some objections to the possibility of free will are dealt with by pointing out the prima facie *evidence and showing how an assumption that everything must be either deterministic or random can result from an unsupported assumption of universalism.*

Anyone who considers arithmetic methods of producing random digits is, of course, in a state of sin. (John von Neuman[1])

The demonstration that no possible combination of known substances, known forms of machinery and known forms of force, can be united in a practical machine by which man shall fly long distances through the air, seems to the writer as complete as it is possible for the demonstration of any physical fact to be. (Simon Newcomb, Professor of Mathematics, John Hopkins University, 1901)

Introduction

In this chapter, I wish to show how free will might be brought about and how this fits into the developmental, social, and evolutionary context of the entities that most clearly exhibit this ability: adult humans. That is, I start from the ability that developed humans seem to have and consider how it might be brought about.

In a way, I would like to say that there is a *mechanism* for free will, but our archetypal pictures of mechanisms and free will are so inimical to each other that juxtaposing them almost forces a choice between them. That is to say, it *seems* we must choose between either (A) that there is free will, but this is not implemented in any mechanism, for it is the nature of mechanisms that they are predictable; or (B) that there is no free will, because whatever there is must be implemented in a mechanism of some kind. I will argue that we *can* have both free will and a mechanism to implement it. I perform this trick by arguing that our picture of mechanisms is inadequate and that we make the mistake of confusing what we can model or understand and the possibilities inherent in what we are trying to model. In particular, I will argue that while modelling the world using either deterministic or (effectively) random processes is common, this does not mean that the world is so composed.

For most of this chapter, I will assume that free will is possible, and thus that it is somehow brought about. The question that I will focus on is *how* it might be brought about. It is, of course, always possible for someone to simply *assert* that free will is impossible. It is my experience in discussing implementing free will that there are some people who just cannot abide such a project, because it is part of their worldview that free will *must be* impossible—if you are one of these people, I suggest that you read the last section, which deals with such philosophical arguments, before reading the rest. If you are prepared to conceive that free will might exist, then I suggest that you read the chapter in the order it is presented (of course, the latter kind of person has a *choice*).

Modelling and Context

All usable modelling is context-dependent. That is to say that a model (or theory) will have a *scope*, which is the set of circumstances under which the model works.[2] This scope is related to the contexts within which the model was developed and validated (though not necessarily restricted to this). Even in physics, there is (as yet) no *universal* theory: quantum physics holds for microscopic events, Newtonian physics for events at an everyday scale, etc.[3] Even when a theory is said to hold universally *in theory*, the conditions under which it is *usefully* applicable do not always pertain. Thus, if you are in the natural world with no instruments or tools, then Einstein's theory of relativity will not help you understand or predict your environment. For more mundane models such as the ones that capture the movement of an unforced, frictionless pendulum or the process of protein manufacture in the cell, the relevant context is far more obvious (though not necessarily made explicit).

When something interferes from outside the modelling context, we often use a *proxy* for this *noise* in the form of an effectively random input (for example, a *pseudo-random* generator). This is often the best we can do, because we cannot extend the model to capture what is beyond the modelling context, but a random source at least mimics the extra-contextuality of this interference. Thus, from within the context, causal factors are often either encodable as explicit parts of a model or represented as random. However, this does not make this interfering cause random in any other sense (i.e., in a different or wider context). Such a modelling approach might well result in it being represented, from the point of view of an exterior context, by a combination of determinism or randomness. However, this does not mean that this is the nature of any mechanism within the brain, i.e., in the interior context. From the exterior context, the functional properties of free will are important, because this explains why free will might be useful (and, hence, why it might have evolved). Thus, I next look at some of these functional properties before turning to a process that might occur within the interior context in later sections.

A Functional Description of Free Will

Free will is more evident in some circumstances than others. It has, presumably, developed in our species as the species evolved (free will is something that distinguishes us from, for example, unicellular organisms). Thus, it is likely that free will has given us some selective advantage, otherwise it is hard to see why it would have arisen.[4] I suggest that the properties of free will that are relevant because they have the potential to provide such selective advantage include the following:

1. **Exterior unpredictability.** From the point of view of a competitor, the actions of the individual possessing free will are (at least somewhat) unpredictable.
2. **Interior rationality.** One's actions lead to one's goals. That is, from an internal view, the actions are consistent with trying to achieve the goals and tend to work toward achieving these goals.
3. **Social accountability.** When requested, the individual can produce an explanation for the (previously somewhat unpredictable) action in terms that establish its rationality to others. That is, a public account of one's decision process can be made so that it can be seen how one's actions were related to one's goals.

The advantages of these arise (on the whole) in social contexts, but it is in social contexts that humans predominantly exist. These social contexts are unavoidable, because the survival of humans seems to come from their ability to inhabit many different ecological niches due to their social adaptivity (Reader, 1990).

In a partially competitive social situation (where it would be to your competitors' advantage to guess what you will do), there is obvious advantage in not being completely predictable. At the same time, one needs to perform actions that will further one's own goals. Properties (1) and (2) are trivially easy to reconcile in the special case where the rational thing to do is something effectively random. Take, for example, the direction an animal may bolt when startled. However, this is not the case with most human actions that need to be (and are) far more structured.

Membership of many human social groups and institutions (in the widest sense) is often conditional on being able to demonstrate that one is rational (from the viewpoint of the others in that group) so as to provide some assurance that you will abide by the norms and rules of the group. The reason for this is that incentives and sanctions on the way you behave will probably have some sway over you. You do not let a mad person into your home, not because they are more likely to be more violent than a sane person, but because any of the usual norms and sanctions one might use to constrain behavior may have no effect. For example, shaming such a person out of acting dangerously may not work.

Thus, we see that simultaneously possessing abilities (1), (2), and (3) is potentially advantageous for us humans (and to a lesser degree other highly social animals). If you lacked (1), you might be predicted and, hence, outcompeted; if you lacked (2), you would be unlikely to achieve any of your goals; and if you lacked (3), you would probably be excluded from many social situations that would otherwise benefit you. Of course, this is a little circular, for many human social structures depend upon the fact that we have properties (3) and (2) and would not be needed at all if it were not for (1). This is unsurprising, as this may well have occurred as a result of the coevolution of our social structures and abilities, as has been suggested for language (Deacon, 1992).

Criteria (1), (2), and (3) form our requirements for an implementation of free will. It is notable that these criteria are each about different contexts: (1) concerns only the external viewpoint of a competitor; (2) is only about the internal, cognitive context; and (3) concerns the translation of the internal into the external context by an individual.

A Proposed Mechanism for Free Will

The key question is: How can these requirements be reconciled via mechanisms that might occur in a brain? We get a clue from human ontogeny. Free will emerges during development. (A human adult has free will, while a single-celled foetus does not in any meaningful sense.) This is not an instantaneous, all-or-nothing ability but one that develops in us over time. As we grow up, it becomes helpful to have created an internal context that is insulated from outside inspection so that others cannot predict what one is going to do, but is sufficiently coherent with others so that social presentations of the content of the internal contexts are judged as reasonable by others.

I suggest that the brain has evolved over millions of years so as to facilitate the development of free will as it develops into adulthood. Thus, the proposal is to implement free will using a suitable developmental process. The type of process I have chosen is of an evolutionary nature—it is the suggestion that biological evolution has resulted in the ability to maintain another evolutionary process in the brain[5]—a process that results in the mature brain being able to direct behavior to the advantage of the individual. The suitability of an evolutionary learning process as the engine of free will is suggested by the properties of such processes as established in the field of Evolutionary Computation, in particular, of Genetic Programming (GP). First, GP is a creative technique, often coming up with unexpected solutions (Koza et al., 1999). Second, the mechanism of sexual recombination in GP acts to maintain the maximum variety in the population as it learns, because it preserves the subtrees but tries different combinations (Koza, 1992, 1994). Third, within the space of solutions, there is a large set of different solutions that will result in the same behavior within a particular training context but will diverge arbitrarily in other contexts (Gathercole, 1998). Fourth, it comes up with solutions that match the goal, which is implicit in the selection mechanism (Koza, 1992, 1994). Last, there is some evidence that it facilitates the evolution of evolvability (Altenberg, 1994).

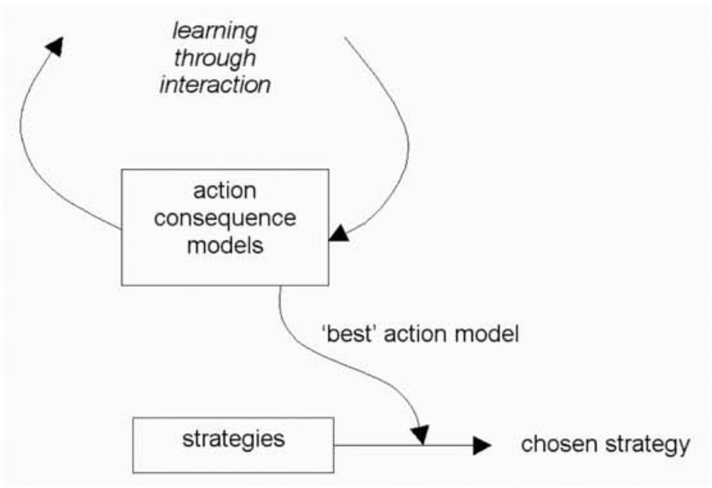
I suggest adapting mechanisms from GP to produce a model of the mind that meets the three criteria (1), (2), and (3). The proposed mechanism is as follows:

- There is a *space* of possible strategies that are constructible from a *language* of steps, conditionals, and actions—this could be formal language as in GP but does not need to be.
- There is a current *population* of strategies from that space that are being evolved as the result of experimental variation of these and their evaluation (based upon the success of using the strategies or similar strategies)—one can think of these as representing the range of alternatives that one considers in making a decision.
- The language of these and many of the original archetypes for these strategies have a social origin, i.e., that primitives, the modes of their combination, and their meanings are socially shared.
- The language must be suitably *open-ended*—that is, strategies that are similar in terms of effect must be expressible in many different ways, and there must not be a hard upper bound on their complexity.
- The success of strategies will be according to their effectiveness in social situations. This does not require the use of some artificial and one-dimensional fitness function, but strategies can be reproduced (or discarded) as and when they are successful (or fail).

This basic structure is augmented in two ways. First, the ability to anticipate the results of strategies is added, thus allowing for the evaluation of strategies by whether they produced the expected results as well as by the extent to which they furthered goals. Second, the (limited) coevolution of the evolutionary operators is allowed. Thus, there are the following additional components:

- The strategies are associated with anticipations of their effects, so that they can be evaluated in terms of whether they produced the anticipated effect as well as their effectiveness in furthering the goals of the agent. This is illustrated in Figure 1. For more on algorithms that implement and use such anticipation (see Butz et al., 2003).
- The operators that act upon the population of strategies and their anticipations to produce new variations are evolved—the operators act upon

Figure 1. Adding anticipation to the development of strategies



themselves to produce variations of themselves, and they are selected (at least partially) according to the extent to which they maintain the unpredictability of these strategies. This is illustrated in Figure 2. An example of these sorts of algorithms and their properties are discussed in Spector (1996) and Edmonds (2001c).

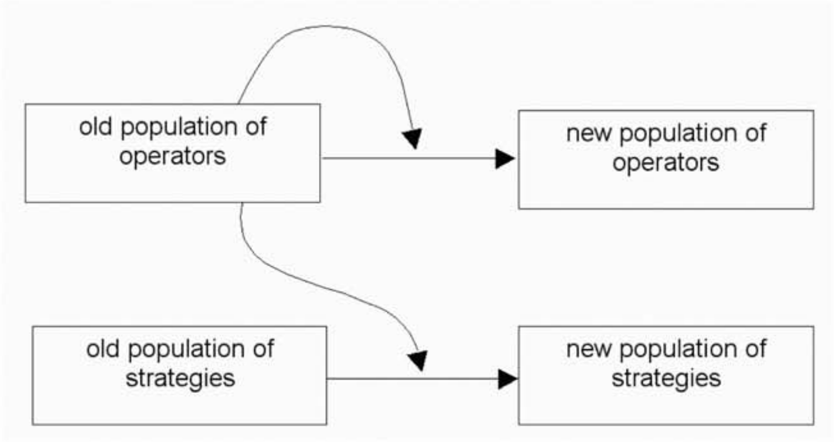
Obviously, in the human case, this is, at best, a mere simulaculum of the complexity of human thought and development. For example, it is becoming increasingly clear that the human brain is fundamentally context-dependent in its processing (Kokinov & Grinberg, 2001). It would be possible to enrich the above outline to include all of these—for example, by including elements such as those described in Edmonds (2001b). However, this is beside the point of this chapter. I do not claim that the detail of the above is what happens in the human case, although I do suggest that some evolutionary process like it might be occurring.

These suggested mechanisms, when used in a socially embedded individual, will satisfy my criteria. I review each in turn.

Exterior Unpredictability

From the point of view of someone else, what is observable is the behavior and reports of other actors. From these observations, one may attempt to infer (or, more accurately, guess) the strategy that the individual was attempting to pursue

Figure 2. Co-evolving the operators of variation


old population of operators
new population of operators
old population of strategies
new population of strategies

and from that predict the future behavior. However, if there are many different strategies that will produce the same behavior as that observed, then when the situation changes, the difference between the guessed underlying strategy and the actual strategy may result in different behaviors being exhibited than those predicted from the inferred strategy. The sharpness of the drop in predictive accuracy with a change in context depends upon the algorithms used for the development of the internal strategies and for inferring other's strategies. This context-dependency is especially sharp in GP. In Jannik (1994), it is shown how GP can be used to produce random sequences by coevolving two populations of programs: the first population selected dependent on success at *not being predicted* by the best of the second population, and the second selected on the basis of success at predicting the first. The result is a modelling *arms race*, with the two populations constantly evolving. This is similar to what happens in cases where agents with GP-based rationality compete in the same task (Edmonds, 1999b).

Interior Rationality

That an individual should act so as to further his or her own goals is unsurprising. However, it is more difficult to see how this can be maintained in the presence of the drive to (exterior) unpredictability implicit in the account above. The answer comes from the GP structure. There are many different strategies that will result in the same behavior in any defined set of circumstances. This comes from the redundancy and open-endedness of the language of strategy expression. All of these strategies can be rational in the sense of furthering the individual's goals. However, in new circumstances (i.e., those not in the defined set), these different variations might well result in very different behavior. Further new developed variations on these different strategies might also result in very different behavior, even back within the established set of circumstances. This *sharp context-dependency* in strategy is a consequence of the GP learning algorithm (Altenberg, 1995).

Social Accountability

The ability to produce an account of why one took any particular action that establishes to the satisfaction of one's peers that one's actions were rational comes out of the shared language of strategies and the underlying rationality of the chosen strategy. Clearly in the human case, the formation of the self includes significant social elements. Elsewhere (Edmonds, in press), I argue that we use our models of others that we infer from our observation of them as a basis for our own self-models, and vice versa. If this is the case, this would provide a deeply shared basis for action strategies. However, this is not necessary for these proposals. Here it is sufficient that the internal strategies can be expressed in a shared language when restricted to the relevant social context and are coevolved.

This proposal also answers the objection that the existence of free will presupposes an infinite regression back in time. That is, free will is only possible if the decision mechanism and the previous state are freely chosen, which, in turn, is only possible if the mechanism and the previous state before that is freely chosen, etc.[6] However, the suggestion that free will evolves in the brain as a human grows up meets this objection—free will emerges by a sort of bootstrapping process in a way that is analogous to how life developed. In such an evolutionary process, if you try and chase any particular decision backwards in time, then you merely increase the difficulty of modelling it, so that this becomes impractical (Edmonds, 1999a). The roots of decisions are lost back in the evolutionary process. In a sense, this process can be seen as a way of amplifying the difficulty of modelling (from an external point of view) from small difficulties to insurmountable ones. A rather misleading way of saying this is that infinitesimal amounts of free will are amplified up to effective amounts by the evolutionary process (just as tenuous and primitive forms of life that are barely distinguishable from mere chemistry have developed into the plethora of life forms we know of today). The strangeness of this amplification from the infinitesimal way of expressing the process results from the attempt to impose a context-independent account on a process that effectively creates a new context—free will is not fundamentally a matter of degree, just as life is not a matter of degree.

Although many parts of this proposal have been implemented (see above references), and I can see no reason why this cannot be implemented, there are still practical difficulties in doing so. These are not so much in the programming required, although no doubt there are several such lurking for a programmer to discover, but more in the environment necessary for such an ability to become manifest. This is because of the social embedding of the free will ability. As I have conceived it (and am convinced it must be like), it requires a complex society of peers for it to function. This is directly analogous to our linguistic ability. Yes, it needs some hard-wired mechanisms in the brain, but it also requires a society for it to develop. Thus, to test this proposal, it would be necessary not only to program one agent as suggested (which is, I claim, feasible) but to allow it to develop inside a society of other such agents. There seem to me to be two ways of doing this: allowing it to develop in interaction with humans or coevolving a society of software agents all with this ability. In either case, the length of the debugging cycle is prohibitive. Thus, testing this proposal must wait until there are routinely societies of interacting software agents.

Social and Cognitive Views

This picture of the relationship between the human mind and its social context as a partial explanation for human intelligence is the core of the *social intelligence* (Kummer et al., 1997) and *Machiavellian Intelligence* (Byrne & White, 1988) hypotheses. In particular, the latter version is almost an inevitable consequence of the processes I suggested above. The dual needs of making one's actions unpredictable and continuing to further one's own ends, effectively involve masking one's true intentions. (A Machiavellian without such qualities would not be very successful!) Yet criterion (3) softens this, because it requires that any such action must be justifiable in a socially acceptable way. Thus, the above account of free will is that it would occur and have meaning in a social context.

That the brain implements such a process (or something equivalent) is not clear. That it could implement such a process is indicated by the discovery that it utilizes evolutionary processes as part of its functioning (Edelman, 1992).

Some Other Views

Most other views that entertain the possibility of free will are *compatibilist*. That is to say, they see free will as compatible with determinism. Many of these seem to take this position due to their simultaneous commitment to determinism and experience of free will. My view is that determinism is simply false, for it presupposes that models can always be expanded to include extracontextual interference as fixed rules inside the model (see the section above on context-dependency and modelling).

Sloman (1993) and McCarthy (2002) both argued that it is sufficient that an agent is able to consider and decide what it will do without undue exterior constraint. This is part of the picture but would be inadequate if the behavior of that agent was predictable by others. If you could guess or infer how that agent reasoned and what factors it based its reasoning on, then, from your point of view (a view exterior to the agent), you might be able to engineer the circumstances of that agent so as to influence the choice it came to. In this case, its choice would not be free, despite the fact that it deliberated and came to its choice itself. Thus, being able to deliberate without undue constraint is insufficient for free will as most people understand it.

Dennett (1984) also argued in this way but pointed out the basic irrelevance of strong versions of free will, arguing that all important aspects of free will, from the point of view of the actor concerned, are possible. Thus, Dennett managed to sidestep the issue of context-dependency, allowing the philosophy to cling to its supposed universality, but at the cost of irrelevance to any of the real[7] issues concerning human choice. Thus, he misses the role that the mental evolutionary process has in separating internal and external contexts and how this essential context-dependency lies at the root of the phenomenon of free will.

McCarthy (2002) also characterized free will as a matter of degree, roughly corresponding to the lack of exterior constraint upon an agent. This is a useful descriptive shortcut but does not correspond in any meaningful way to the interior process that implements the ability. There is no amount of complexity required for free will to be possible, rather there is a set of basic abilities and an appropriate social environment. Again, the analogy with life is striking. Life is not a matter of degree — there is no number that specifies how *alive* something is. However, it requires basic abilities (reproduction, etc.) and an appropriate environment. Of course, there are lots of possible forms that life can take, and I would expect that free will, being a creative ability, also takes lots of different forms.

Philosophical Objections to the Existence of Free Will

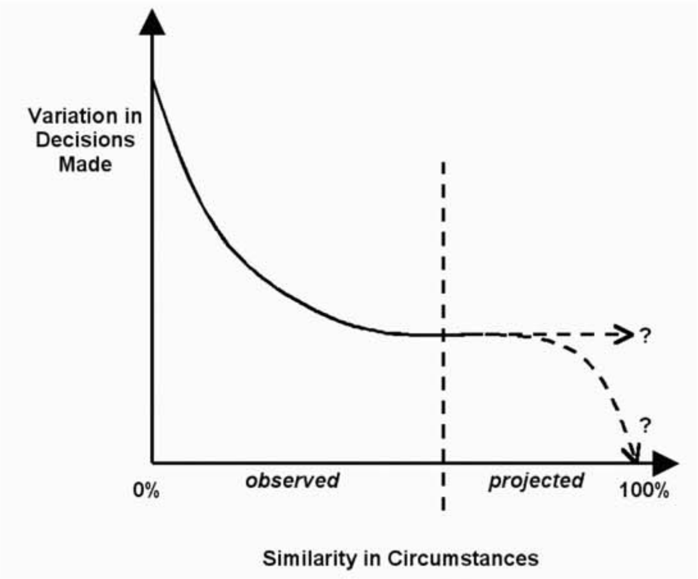
Such a proposal as this inevitably provokes many philosophical responses. They are mostly variants of an *a priori* conviction that free will is impossible, and so my suggestion must be inadequate.

The most easily dismissed of these comes out of an assumption that the world (and, hence, human decision making) is deterministic, despite the fact that evidence does not support this. The spectacular success of quantum physics tells us that, at least at the atomic and subatomic levels, matter is fundamentally nondeterministic. A slightly more reasonable assumption is that macroscopic events are, in practice, deterministic. That is to say, for large ensembles of atoms (for example, humans) at any particular point in time, the more similar the past situation of this ensemble, the more similar the future will be. This is not a practical argument for the human case, because the degree to which the circumstances must be identical is far beyond what can ever be arranged. These circumstances must include the whole memory of the individual, and all of the individual's past circumstances, thoughts, and memories. Thus, this convergence of human choice with convergence in circumstances is a theoretical assumption (via analogy with simple systems like billiard balls) rather than a matter of evidence.

The extent of the evidence and the possible outcomes in the limiting case of complete convergence of past circumstances is illustrated in Figure 3. The line in this figure is a hypothetical illustration of how the variance of human decisions (in some particular, measurable case) might change due to the variance in circumstances. The purpose of this graph is to try and make clear the underlying pictures that people have about determinism and free will and thus highlight how unsupported the assumption of human determinism is. The graph illustrates the fact that as circumstance becomes more similar, humans do not (beyond a certain degree) always come to the same decisions. The convinced determinist will insist that this is due to the small variation in circumstances that remains and, furthermore (despite the fact that there is no indication or trend which suggests this), if the circumstances (including memories, etc.) were identical, then an identical decision would result. This amounts to no more than the assumption of determinism in a different form.

A more sophisticated response is that the world is either deterministic or random, and so human decisions must be the same. However, as I argue at the beginning, this is a result of the context-dependency of our modelling. We use randomness as a model of what we cannot model, and we impute this upon the parts of human

Figure 3. An illustration of the supposed convergence of behavior with the similarity of situation



decision making we cannot model. That we cannot model it is unsurprising, because this is (part of) the purpose of free will—to separate the modelling from internal and external points of view. The fact is, there is simply no evidence that human behavior is either rigidly determined or random in practice; there are many indications to the contrary. Hence, the insistence that human behavior is like this is more in the nature of a theoretical commitment. The fact that humans (on the whole) can produce a credible reason for their actions (after the fact) makes attributing randomness as a significant cause difficult to maintain, and yet there are many cases in which humans retain the facility to surprise. For example, huge effort has gone into predicting stock market price changes (which result directly from human decisions) without success, and yet the evidence is that these are far from random.

Some (for example, many economists) will assert that however the individual decides on action, taken *en masse*, individual actions are, in effect, random. Even if this was the case (and it is not[8]), this would not mean that an individual's action was random, merely arbitrary and uncoordinated with others' actions.

There is the objection to free will mentioned above, that free will presupposes prior free will. That is to say, for free will to be possible now, it must have been possible previously, as the present choice is based upon past memories and the existing decision mechanism. If these are not free, then the present choice cannot be free. The conclusion from this observation is often that free will is impossible. However, the proposed mechanisms directly deal with this. This argument does not rule out the possibility of free will, any more than a similar argument would rule out the possibility of life. Rather it shows that there is some recursion here, just as life is involved in determining and producing new life, so it is possible that free will is involved in determining and producing more free will. Just because life presupposes prior life to beget it in the process of reproduction, this does not mean that life does not exist. It was this thought that first suggested an evolutionary process to me.

While not being simply a matter of degree, there is ultimately no hard and fast boundary between having free will and not having it. Like intelligence, it is a complex ability, which may be compared using numerical measures as a sort of sloppy shorthand (A is more intelligent than B). Most people do not have any problem with the idea that intelligence develops with the individual. This bothers some thinkers who wish to insist that it is an all-or-nothing property, without evidence that this is the case, but more, I would guess, on the grounds that it enables them to create arguments such as the recursive one above. Hofstadter (1985) put it nicely when he said:

> *Perhaps the problem is the seeming need that people have of making black-and-white cut-offs when it comes to certain mysterious phenomena, such as life and consciousness. People seem to want there to be an absolute threshold between the living and the nonliving, and between the thinking and the "merely mechanical,..."*

This brings us to the most fundamental difficulty: it is part of the nature of philosophy to seek universal (non-context-dependent) descriptions of the world. This universalism is partly due to the widespread and accepted practice of using counterexamples in philosophical argument. These counterexamples are considered as philosophically valid, however weird and extreme they are, and this leaves any context-dependent proposal vulnerable. This tendency leads, in practice, to an assumption that such a universal view is possible once the details of particular contexts are filtered out. If the mechanisms of free will are developed in part for their effectiveness at separating internal and external contexts, then its existence and the applicability of a philosophical approach are opposed (to the extent philosophy is limited to potentially non-context-dependent arguments or truths). One is then left with a choice: accept that such free will can exist, for which there is some evidence (albeit mostly anecdotal), or rely on the universality of philosophy (which is pure assumption, because there cannot be evidence for this).

Acknowledgments

Thanks to the participants of the AISB symposium on "How to Design a Functioning Mind" in Birmingham in April 2000 for their discussion and comments on the first version of this chapter (Edmonds, 2000a), as well as Aaron Sloman and colleagues at Birmingham University with whom I discussed some of these ideas.

References

Altenberg, L. (1994). The evolution of evolvability in genetic programming. In E. K. Kenneth (Ed.), *Advances in genetic programming* (pp. 47–74). Cambridge, MA: MIT Press.

Altenberg, L. (1995). Genome growth and the evolution of the genotype–phenotype map. In W. Banzhaf & F. H. Eeckman (Eds.), *Evolution as a computational process* (pp. 205–259). Berlin: Springer-Verlag.

Butz, V. B., Sigaud, O., & Gérard, P. (Eds.). (2003). *Anticipatory behaviour in adaptive learning systems*. Berlin: Springer, Lecture Notes in Artificial Intelligence, 2684.

Byrne, R. W., & Whiten, A. (Eds.). (1988). *Machiavellian intelligence: Social expertise and the evolution of intellect in monkeys, apes, and humans*. Oxford: Clarendon Press.

Castelfranchi, C. (1995). Guarantees for autonomy in cognitive agent architecture. In M. Wooldridge & N. R. Jennings (Eds.), *Intelligent agents: Theories, architectures, and languages* (pp. 56–70). Berlin: Springer-Verlag, Lecture Notes in Artificial Intelligence 890.

Deacon, T. (1992). *Brain–language coevolution*. Reading, MA: Addison-Wesley.

Dennett, D. C. (1984). *Elbow room: Varieties of free will worth having*. Oxford: Oxford University Press.

Edelman, G. M. (1992). *Bright air, bright fire: On the matter of mind*. London: Penguin.

Edmonds, B. (1999a). Capturing social embeddedness: A constructivist approach. *Artificial Behavior, 7*(3/4), 323–348.

Edmonds, B. (1999b). Gossip, sexual recombination and the El Farol bar: Modelling the emergence of heterogeneity. *Journal of Artificial Societies* *and Social Simulation, 2*(3). Retrieved from *http://www.soc.surrey.ac.uk/JASSS/2/3/2.html*

Edmonds, B. (2000a). Towards implementing free will. *AISB 2000 symposium on "How to Design a Functioning Mind,"* Birmingham, April 2000. Retrieved from *http://cfpm.org/cpmrep57.html*

Edmonds, B. (2001b). Learning appropriate contexts. In V. Akman, P. Bouquet, R. Thomason, & R. Young (Eds.), *Modelling and using context — CONTEXT 2001* (pp. 143–155). Dundee, July, 2001. Berlin: Springer-Verlag, Lecture Notes in Artificial Intelligence, 2116.

Edmonds, B. (2001c). Meta-genetic programming: Co-evolving the operators of variation. *ELECTRIK, 9*, 13–29.

Edmonds, B. (in press). The social embedding of intelligence — Towards producing a machine that could pass the Turing Test. In G. Peters & R. Epstein (Eds.), *The Turing Test sourcebook: Philosophical and methodological issues in the quest for the thinking computer*. Dordrecht: Kluwer.

Gathercole, C. (1998). *An investigation of supervised learning in genetic programming*. PhD thesis, University of Edinburgh. Available online at *ftp://ftp.dai.ed.ac.uk/pub/daidb/papers/pt9810.ps.gz*

Hofstadter, D. R. (1985). Analogies and roles in human and machine thinking. In D. R. Hofstadter, *Metamagical themas*. New York: Basic Books.

Jannink, J. (1994). Cracking and co-evolving random generators. In K. E. Kinnear (Ed.), *Advances in genetic programming* (pp. 425–444). Cambridge, MA: MIT Press.

Kokinov, B., & Grinberg, M. (2001). Simulating context effects in problem solving with AMBR. In V. Akman, P. Bouquet, R. Thomason, & R. A. Young (Eds.), *Modelling and using context* (pp. 221–234). Springer-Verlag, Lecture Notes in Artificial Intelligence, 2116.

Koza, J. R. (1992). *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, MA: MIT Press.

Koza, J. R. (1994). *Genetic programming II: Automatic discovery of reusable programs.* Cambridge, MA: MIT Press.

Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1999). Genetic programming: Biologically inspired computation that creatively solves non-trivial problems. In L. Landweber, E. Winfree, & R. Lipton (Eds.), *Proceedings of DIMACS Workshop on Evolution as Computation* (pp. 15–44). Berlin: Springer-Verlag.

Kummer, H., Daston, L., Gigerenzer, G., & Silk, J. (1997). The social intelligence hypothesis. In P. Weingart, S. D. Mitchell, P. J. Richerson, & S. Maasen

(Eds.), *Human by nature: Between biology and the social sciences* (pp. 157–179). Hillsdale, NJ: Lawrence Erlbaum Associates.

McCarthy, J. (2002). *Deterministic free will.* Available online at *http://www-formal.stanford.edu/jmc/freewill2/freewill2.html*

Reader, J. (1988). *Man on earth*. London: Collins.

Sloman, A. (1992). How to dispose of the free-will issue. *AISB Quarterly, 82*, 31–32.

Spector, L. (1996). Simultaneous evolution of programs and their control structures. In P. Angeline, & K. Kinnear (Eds.), *Advances in genetic programming 2*. Cambridge, MA: MIT Press.

Endnotes

[1] Reportedly said by von Neuman at a conference on Monte Carlo methods in 1951.

[2] Of course, what works depends on how useful the model is in furthering one's goals for it, so the nature of a model's scope is more complicated than I indicate here.

[3] This is somewhat of a simplification, for quantum effects can have consequences in the macroscopic world, etc. However, it is true that each theory has its own nonuniversal domain of applicability.

[4] It is possible that free will might be merely a side-effect of another ability that evolved due to the selective advantage it gave. However, because free will seems so costly in terms of the time it takes us to make up our minds, this seems unlikely.

[5] This is supported by the fact that processes of an evolutionary nature seem to be occurring in the brain (Edelman, 1992).

[6] Thus, having free will can be seen as the ultimate extension of autonomy, where this is conceived as the ability to change, select, and adopt goals (Castelfranchi, 1995).

[7] Dennett, being a good philosopher, does not, of course, use the word "real" in this sense but rather means "practical," "relevant," etc.

[8] Which it is not, for a consequence of this would be that in larger social systems, the randomness would tend to cancel out as a proportion of the total system by the law of large numbers, and the system become more predictable. This is not supported by observation of such systems.

Chapter 7

Images of Mind: In Memory of Donald Broadbent and Allen Newell

John Fox
Cancer Research UK, London, UK

Abstract

The idea of "mind" did not spring fully formed into human consciousness. On the contrary, it has been articulated slowly through the millennia, drawing upon countless metaphors and images in different cultures at different times. In the last 50 years, the concepts of conventional science and technology have provided the primary images that we employ in the discussion of mental processes. Unfortunately, there are many competing perspectives, each of which is incomplete when it comes to explaining mental phenomena, and most are inconsistent. In this chapter, four prominent images that have influenced cognitive science in the last half-century are considered. These distinguish structure *and* process *theories of mind developed in psychology, from the* epistemic *and "*pathic*" theories of mental states emerging in artificial intelligence (AI). A critical challenge is to construct a theory of cognition in which these different images of mind can be seen to be complementary views of a single system. The chapter closes with an example of such a theory.*

Introduction

The various properties that distinguish Mind from what is not Mind are summed up in three great general properties, named Emotion or Feeling, Volition or Will, and Intellect or Thought. Chambers' Information, 1884

It is not clear when the idea of "mind" emerged. Early "mentalistic" terms were originally used to refer to physical body parts in ancient Hebrew but gradually acquired additional associations. For example, the most primitive meaning of the Hebrew word *nepesh* was throat or gullet, which came to be associated with the wish for food and drink, later a more general notion of "life force," and in due course, it came to refer to the "self." Similarly, the word *ruach* initially referred to breath or the organ of breathing and later the power or force behind the breath, and hence motivating power. Finally, *leb* refers to the heart. The notion of the heart as the seat of human emotions emerged gradually from that of the heart as the chest or bosom, but according to MacDonald (2003) it came to function in all aspects of human existence (emotional, cognitive, and volitional). Its most important Hebrew usage is "*in passages that clearly indicate intellectual, cognitive and reflective operations*."

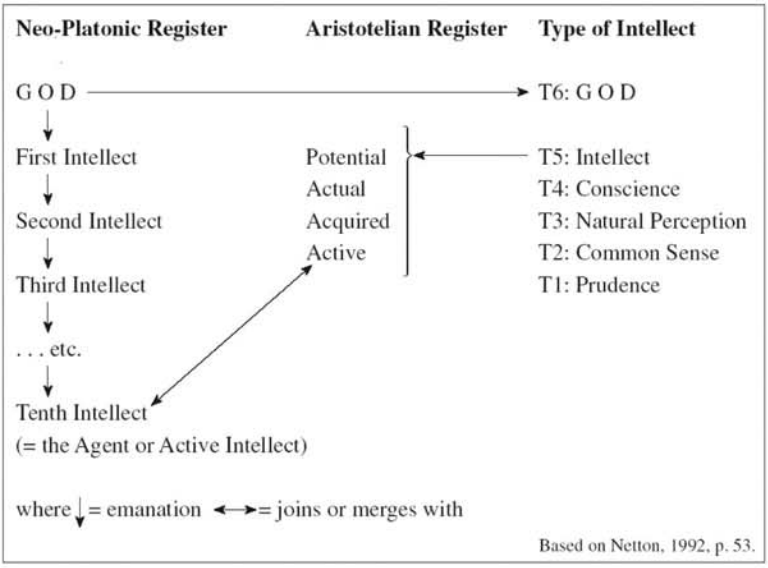
Greek learning was notable for the systematization of knowledge, as in the development of technical disciplines like arithmetic, geometry, astronomy, music, optics, medicine, logic, and politics. Psychological thinking, however, does not appear to have developed far, though Aristotle and Plato made a start. Aristotle tried to understand the "intellect" with such distinctions as the *passive intellect*, which "receives intelligible species," and the *agent intellect*, which "produces intelligible objects"; notions that are only just intelligible to the modern reader. Plato and other Greek thinkers expressed other intuitions, some of which we can still find in contemporary "folk" psychology, though there were strong dissonances as well. For example, "one of Plato's main ideas was that the rational soul (i.e., mind) is immortal, which is not congruent with a modern, materialist point-of-view."[1] Generally, it seems that these early writers were struggling to understand mental phenomena, as we still do today.

One obvious reason for the difficulties faced by early thinkers in understanding mind in anything like modern terms was the existence of competing philosophical modes of enquiry and ideologies. Religion, and attempts to explain all existence with religious ideas, strikes the secular scientific investigator as an obvious source of confusion. Distinctions between mind as a cognitive entity and mind as coextensive with "soul" were constantly blurred. Christian thought appears to have embodied a curiosity about the subjective experience of cognition but for a long time could not shake off the need to ensure its consistency with religious doctrine. By the ninth century, Islamic scholars had greatly developed the Aristotelian and other earlier systems toward a differentiated view of cognition that distinguished perceptual, epistemic, rational, intentional, ethical, and other aspects of the intellect. However, this was all set within a framework bounded by the unknowable mind of God (as in Afarabi's model of mind shown in Figure 1).

The separation of questions to do with the nature of rational mind and more mystical ideas was slow and incremental, though we tend to see history in terms of sudden lurches toward modern ideas in "periods," such as the Renaissance and the Enlightenment. At any rate, by the 19th century, philosophical and introspective enquiry had acquired a fairly clear, if intuitive, idea of mind, which could be summarized in terms of a small number of basic ideas, as illustrated by the quote from *Chambers Encyclopaedia* above.

By this time, philosophical introspection was also beginning to give way to modern methods of investigation of mental phenomena (for example, see Solso, 1998, for a historical review), and by the 20th century, a number of scientific paradigms had been established. The medical and physiological tradition sought explanations of mind in, for example, the structure of the brain and neural apparatus. The psychoanalytic traditions were more concerned with experience than physical mechanisms, explaining mental phenomena in terms of abiding human preoccupations and culture-specific metaphors. Behaviorism investi-

Figure 1. An Islamic view of the intellect (reproduced with permission from MacDonald, 2003)



Based on Netton, 1992, p. 53.

gated human and animal behaviour in the laboratory without invoking mentalistic ideas.

These and other streams of thought played out over the first half of the 20th century. By the middle of the century, most investigators took the modern perspective for granted: brains implement minds, and the best way to study their relationship is through laboratory experiment and the development and testing of mechanistic models of structure, function, and behaviour. This is the starting point for the present chapter, the purpose of which is to bring together a number of contemporary paradigms that spring from around that time.

Metaphors for Mind

As in most sciences, the cognitive sciences aim for a theory that unifies its field, in this case from neuroscience to everyday mental experience. That ambition is shared with the scholars of antiquity, but we are more aware of the complexity of what we are taking on. Contemporary cognitive scientists deal with this complexity by using different metaphors and images to explain different aspects of cognition. This strategy is common in the life sciences. Take the case in biology of "how cells work." Plant and animal cells are understood using many distinct metaphors. These include the static "architecture" of the cell which is built up from specialized substructures and organelles; the cell as a dynamic "signalling network" in which lots of little chemical messages whiz from point to point causing things to happen at different places in the cell; the cell as a "bag of chemicals" with concentrations that vary dynamically over time according to quantitative chemical laws; or even as computational systems in which DNA sequences are "programs" for building proteins or controlling the cycle of cell growth, division, and death.

Cognitive scientists trying to understand "how the mind works" have adopted a similar strategy. Unfortunately, in this case, it appears to be leading to fragmentation of the subject, with different communities who work on different aspects of the problem using different languages and concepts and who frequently arrive at positions of mutual incomprehension.

In the next few sections, I review some of the theoretical frameworks that have emerged in the last half-century, starting with psychology and progressively introducing ideas from AI and, most recently, formal theories of knowledge and mental states. I will start with the work of two famous figures, Donald Broadbent and Alan Newell. This choice is arbitrary, in that I wish to remember two of my personal mentors, but they are also important and representative figures in complementary fields of cognitive science. At a time when ideas from neuroscience and from computational science are becoming increasingly successful but also taking different and somewhat inconsistent directions, it may be useful to remember them and the intellectual continuity they represent.

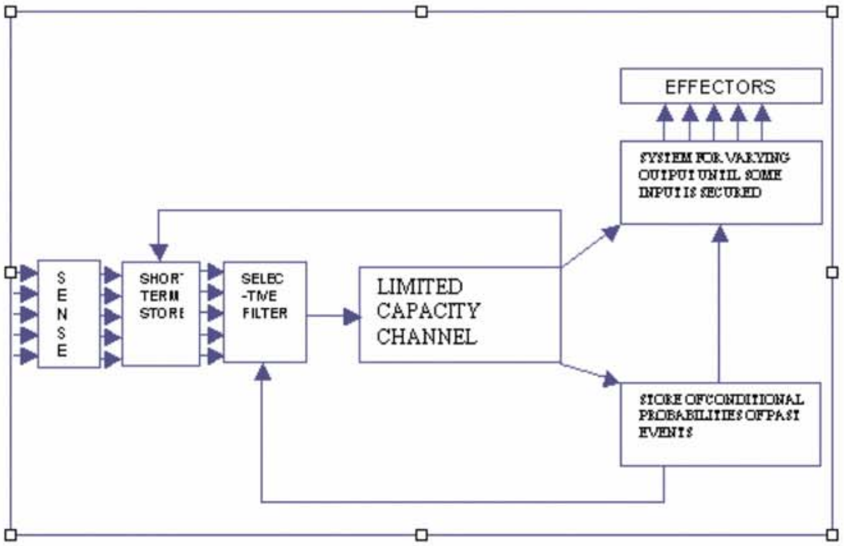
Statics: Minds as Information-Processing Architectures

A nervous systems acts to some extent as a single communication channel, so that it is meaningful to regard it as having a limited capacity. ... incoming information may be held in a temporary store at a stage previous to the limited capacity channel:.... The maximum time of storage possible in this way is of the order of seconds. D. E. Broadbent, Perception and communication*, 1958, pp. 297–298*

Donald Broadbent (1926–1993) was director of the Medical Research Council's Applied Psychology Unit in Cambridge, United Kingdom. He made many contributions concerning the organization and integration of human cognitive functions, like memory, attention, and decision making, drawing on experimental investigations and detailed observations of the complex demands that real-world environments place on human beings. His applied focus gave him a keen awareness that cognitive functions are strongly affected by environmental factors, like high noise levels, and other stressors, and how they affect human physiology. His engineering training also allowed him to see how the information theory that developed during and after the global conflict of 1939–1945 could shed light on these processes. The centerpiece of *Perception and Communication* (1958), the book that established his reputation, visualized functions like memory, attention, and decision making in an image of the mind as an information-processing system made up of a number of interacting physical components (Figure 2).

"Box and arrow" diagrams of the sort that Broadbent used have continued to be popular methods for providing an image of mental processes in psychology, because, under fairly simple interpretation rules, the organization of the components often seems to yield unambiguous, and experimentally testable, predictions about human behaviour. Box and arrow conventions can also be used at different scales, from the detailed processes involved in recognizing words to the high-level organization of cognitive processes thought to be involved in human consciousness (Figure 3).

Figure 2. Broadbent's model of selective attention (reproduction of Figure 7 in Perception and Communication)


EFFECTORS
SYSTEM FOR VARYING OUTPUT UNTIL SOME INPUT IS SECURED
S E N S E
SHORT TERM STORE
SELEC -TIVE FILTER
LIMITED CAPACITY CHANNEL
STORE OF CONDITIONAL PROBABILITIES OF PAST EVENTS

Such schematic diagrams are routinely used to represent and design complex systems in engineering, but in engineering, we often already understand how the bits work and can use standard engineering conventions to convey information unambiguously. Clear conventions of this kind do not really exist in psychology, so the purpose of such diagrams is largely to facilitate discussion of systems that we do not understand in any detail and those for which we cannot even be confident of the basic components. Unfortunately, despite their simplicity, box and arrow diagrams are open to the criticism that they are at best underspecified and at worst misleading.

Richard Cooper of Birkbeck College London observed that it would be valuable to define some standard conventions for modelling cognitive "modules," their properties, and their interrelationships. His suggestion has been put to practical use in the COGENT cognitive modelling system illustrated in Figure 4 (Cooper & Fox, 1998; Cooper, 2002).

Figure 4 shows a different image of Broadbent's (1958) theory of selective attention created using the COGENT modelling system. COGENT provides a set of standard "cognitive components" for modelling information-processing systems, including processing units, memory buffers and knowledge bases, communication lines, and learning mechanisms. The left panel shows the complete model, representing the cognitive system and its communication with the "task environment." The cognitive system is shown in an exploded view in the centre

Figure 3. Box and arrow models of stages of four subsystems or processes hypothesised to be involved in human "awareness" (Shallice, 1988, p. 402), and a detailed information processing model of how single words are processed and recognized in the human auditory system (Morton & Paterson 1980), reproduced in Shallice (1988, p. 92).

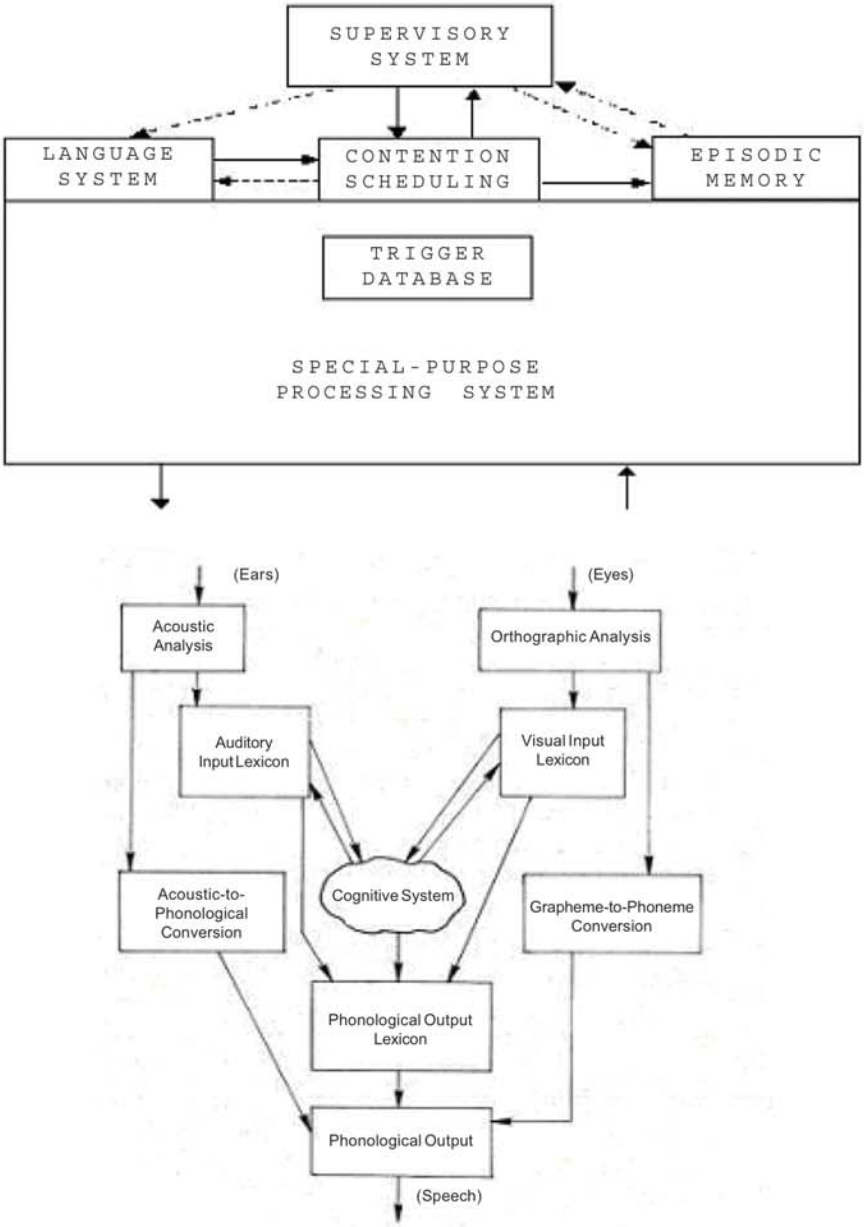
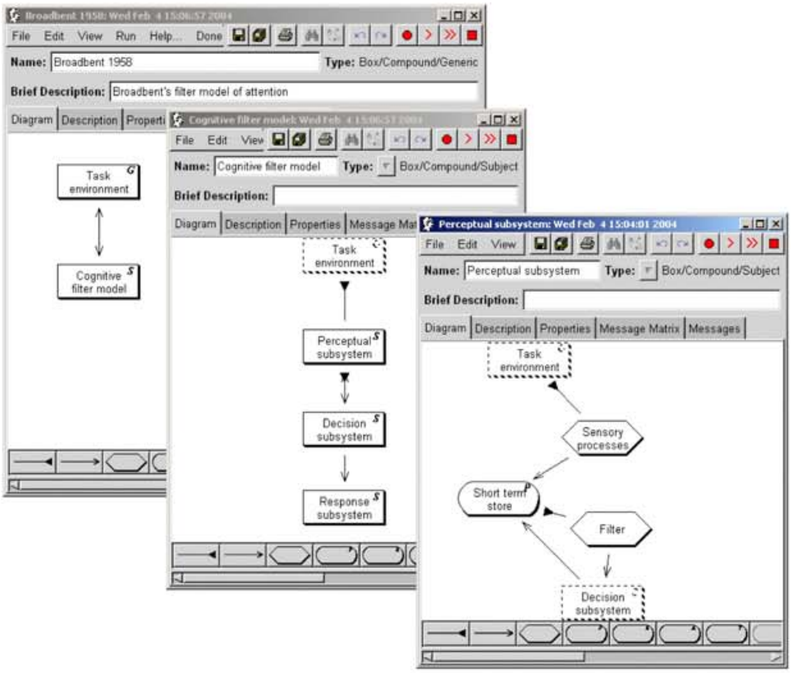


Figure 4. Broadbent's model of selective attention modelled in COGENT (The modeller sketches components of each model compartment using standard components, "exploding" these where needed to define internal structure (see text).)



Name: Broadbent 1958 Type: Box/Compound/Generic

Brief Description: Broadbent's filter model of attention

Name: Cognitive filter model Type: Box/Compound/Subject

Brief Description:

Name: Perceptual subsystem Type: Box/Compound/Subject

Brief Description:

Broadbent 1958: Wed Feb 4 15:06:57 2004
File Edit View Run Help... Done
Diagram Description Properti
Task environment
Cognitive filter model
Cognitive filter model: Wed Feb 4 15:06:57 2004
File Edit View
Diagram Description Properties Message Mat
Task environment
Perceptual subsystem
Decision subsystem
Response subsystem
Perceptual subsystem: Wed Feb 4 15:04:01 2004
File Edit View
Diagram Description Properties Message Matrix Messages
Task environment
Sensory processes
Short term store
Filter
Decision subsystem

panel. It is modelled here as three subsystems, one of which is shown in the panel on the right. The perceptual subsystem is made up of two "processing modules" (shown as hexagons) that are able to operate on information obtained from other parts of the model. In this model, the processing modules can retrieve information from and send information to the short-term store that was an explicit element of Broadbent's model. (In COGENT, processes are viewed as stateless—they do not store information—so memory systems and their properties must be modelled separately.) COGENT supports a range of different types of memory, shown as rectangles with round ends.

COGENT is a flexible tool for describing the static organization of cognitive systems, and it has been used extensively for constructing models of natural systems and designing artificial ones (see Cooper, 2000, and also Glasspool's chapter in this volume). It provides a standard way of visualizing cognitive components, a range of predefined "cognitive modules," and a practical modelling tool. As remarked above, cell biologists use accepted conventions for visualizing many static aspects of cellular processes, such as three-dimensional molecular models and metabolic pathways drawn as flow graphs. COGENT offers clear graphic conventions for describing static cognitive architectures and, as we see in the next section, a precise and unambiguous language for specifying the properties of such systems and hence predicting their behaviour.

Dynamics: Minds as Processes

A single system (mind), produces all aspects of behaviour. ... Even if the mind has parts, modules, components, or whatever, they all mesh together to produce behaviour.... If a theory covers only one part or components, it flirts with trouble from the start. It goes without saying that there are dissociations, independencies, impenetrabilities and modularities. ... but they don't remove the necessity of the theory that provides the total picture and explains the role of the parts. A. Newell, Unified theories of cognition, *1992 (pp. 17–18)*

At about the same time that Broadbent published *Perception and Communication*, the new field of AI was getting going. Two influential conferences took place around this time: the famous 1956 Dartmouth conference, where the field of artificial intelligence acquired its name[2]; and the "Mechanization of thought" conference in Teddington, UK (1957). These meetings brought together many scientists interested in understanding diverse aspects of mind, from vision and pattern recognition to speech, language, and problem solving.

Many in the early AI community thought they were simply developing a new branch of engineering; others felt that their work was shedding light on general principles that underlie intelligence. In the following decade, interest in AI gathered momentum in parallel with the growth of "information-processing psychology." In 1967, the psychologist Ulric Neisser published *Cognitive Psychology*, a book that brought the static images of cognition that Broadbent and others were using together with ideas emerging from work in AI. Neisser believed that computer algorithms might be the key to understanding how static cognitive models could be enhanced to explain the dynamic processes that generate behaviour.

Allen Newell (1927–1992) was a computer scientist who attended the Dartmouth and Teddington conferences. With his collaborator, Herbert Simon, he pioneered the use of computational models in psychology. He was particularly interested in the dynamic aspects of cognitive processes and how they are controlled. His ambition became to develop a "unified theory of cognition" that dealt with both static and dynamic properties. Until his death in 1992, he worked on a series of simple, elegant, and increasingly ambitious systems for modelling mental processes. Through PSG (Newell, 1973) to the OPS series of systems (Forgy & McDermott, 1977) to the SOAR cognitive modelling system that embodied what Newell believed to be a general theory of intelligence (Laird et al., 1987; Newell, 1992), he progressively refined three fundamental propositions:

1. In order to understand mind, we must separate the *fixed architecture* from the *variable component* of cognition. The fixed architecture is the information-processing machinery to be found in all human heads; the variable component is, roughly, what we call knowledge.
2. Newell took the view that understanding knowledge is basic to understanding cognition. His research position was that all knowledge can be represented by simple *condition–action* rules.
3. The *control structure*, the mechanism by which the static cognitive architecture applies knowledge in reasoning, problem solving, decision making, and other cognitive tasks, is a fundamental problem for understanding the nature of a functioning mind.

The fixed architecture that Newell adopted owed much to the model that had been developed by Broadbent and other psychologists. His knowledge representation, the condition–action rule, is a simple yet general construct that can be used for many purposes, from logical operations like deduction to modelling actions and plans.[3] The dynamic control structure that Newell thought was fundamental to flexible cognition is a cyclical process of selecting and firing rules driven by situations and events, the "recognize–act" cycle.

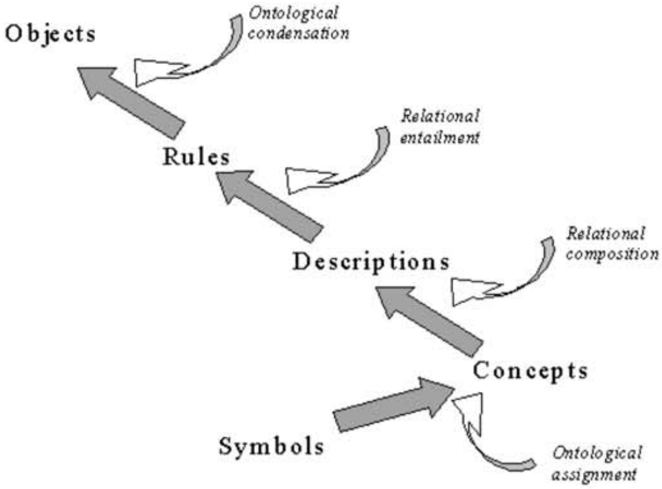
Newell and his many students explored a range of questions raised by trying to understand cognition in this way. Many different schemes for controlling rule execution were explored, culminating in the SOAR system that combined the recognize-act cycle with goal-oriented problem solving and the ability to learn from experience by acquiring new rules. As time has moved on, this particular embodiment of his ideas has become less convincing, as it failed in its original form to explain important phenomena of human cognition (e.g., Cooper & Shallice, 1995). However, Newell's line of work has continued to exert influence on psychologists wanting to understand dynamic aspects of human cognition. Anderson and Lebiere's ACT-R model (1998) and Just, Carpenter, and Varma's 4CAPS (1999) combine production rule theories with strong empirical research programs in cognitive psychology and neuroscience. The direction that Newell initiated continues to provide a promising perspective for understanding the human mind.

During the period that Newell was developing the rule-based approach to cognitive systems, a more formal approach to AI was also emerging, particularly in Europe. *Logic programming* was emerging as another important example of rule-based computation, though here cognition is modelled as a form of logical reasoning. In contrast to the production rules of OPS and SOAR that would fire in response to situations or events, rules in a logic-programming system are evaluated from the point of view of achieving goals. Where production rules lead to a "forward chaining" cycle, logic programming involves a "backward chaining" process of recursive proofs. As in the production rule approach, logic programming sees control as a fundamental aspect of computation. Robert Kowalski, one of the founders of logic programming, viewed all of computation as the controlled interpretation of logical expressions, which he famously summarized as *algorithm = logic + control* (Kowalski, 1979). Logic programming also continues to have an influential place in cognitive science, providing a versatile and expressive AI programming technique with mathematical foundations that are well understood. For this reason, it continues to play an important role in the development of formal theories of intelligent systems (e.g., Das et al., 1997; Wooldridge, 2000; Fox et al., 2003).

Many ideas in cognitive dynamics have been investigated, but rule-based processing possibly represents the most flexible single mechanism for modelling cognitive processes that has been identified to date. Others will no doubt emerge, as cognitive systems probably require different control regimes in different situations. This is evident in current work on "autonomous agents," where it is recognized that agents need to be able to operate both "reactively," in response to events, and "deliberatively," in a purposive manner (Fox et al., 2003).

The COGENT cognitive modelling system described above is not just a tool for creating static images of mental processes; it also incorporates a programming system for constructing and testing models. Cooper (2002) discussed examples of various models, of reasoning, problem solving, decision making, and natural language processing, for example. The programming system COGENT provides is a hybrid logic-programming and production-rule system (Hajnal et al., 1989). Each process in a COGENT model can operate in a reactive or deliberative fashion or in a mixed mode, depending on the function being modelled. The system represents a general tool, not a specific theory of cognition like SOAR or ACT-R. It provides a versatile set of components for modelling cognitive architectures and simulating their behaviour. Furthermore, it incorporates standard component interfaces and may offer a new way in which cognitive scientists can collaborate: independently developing theories of cognitive functions and connecting them together for testing over the Internet.[4]

Figure 5. Knowledge ladder



Epistemics: The Knowledge Level

The trouble with "knowledge" is that we don't know what it is. Donald Broadbent, personal communication, 1975

Newell drew extensively on Broadbent's ideas about the organization of mind, notably the distinction between short-term and long-term memory that Broadbent and his peers did so much to articulate. However, Newell's interpretation was based on computational ideas about the dynamic operation of the cognitive system as well as the functional perspective of different kinds of memory. Furthermore, he believed in the theoretical importance of understanding the *content* of cognition. Where Broadbent viewed long-term knowledge merely as a "store of conditional probabilities" (Figure 2), Newell realized that *if...then...* rules could represent a far greater range of types of knowledge about an agent's task environment.

Broadbent viewed the problem of mind primarily as an engineering challenge, in which the aim is to understand how the cognitive system works and how well it performs (how fast, how reliably) and why it degrades under stress, for instance (Broadbent, 1971). Newell recognized the need for an engineering diagram and an understanding of the components, but he also understood that if psychology is to have anything to say about individual cognition, we need an account of what we know as individuals. Our knowledge, after all, plays the director's role in everything we do. In 1982, he published one of the seminal papers of recent AI, "The knowledge level," which articulated one of the important contributions of AI to cognitive science.

It is now uncontroversial that knowledge can be understood in formal, computational, and even mathematical terms, but also that theories of knowledge require different constructs from those needed for understanding physical or biological systems. In this section, I give a short overview of current knowledge representation theory. To keep the presentation brief, I use the simple image in Figure 5 to explain some of the ideas.

The standard way of talking about knowledge nowadays is to describe it as a collection of expressions in a symbolic language. Such languages can include informal natural language (indeed, I will use such language for my examples), but work in this area is increasingly formal. From an AI point of view, it is necessary to have a formal semantics if we want to design cognitive systems that can apply knowledge in making inferences, solving problems, making decisions, enacting plans, and so on, in a reliable and principled fashion.

A good knowledge modelling language is *compositional*; we build "sentences" in the language out of simpler phrases and sentences according to clear rules. The starting point on the knowledge ladder is the symbol. Symbols are easy to represent in computers, but by themselves, they have no meaning. Meaning is introduced progressively, by defining relationships between symbols using *epistemic conventions*. These conventions sanction the composition of symbolic sentences into increasingly complex expressions in the language.[6]

The most primitive epistemic convention is classification, in which a symbol is assigned to some class (of object or other concept) that it is required to represent. In early computational systems, the only classifications that people were interested in were mathematical types ("integer," "string," etc.). With the rise of cognitive science, however, a further classification convention emerged. *Ontological assignment* assigns a symbol to a conceptual category (usually a humanly intelligible concept, but there is no reason why it must be). For example, the symbol "apple" can be assigned to the class "fruit" using the relationship symbol *a kind of*. Ontological assignment begins the transformation of meaningless symbols into meaningful concepts that can be the objects of cognitive processing. One important cognitive operation that is usually taken to be a direct consequence of ontological assignment, for example, is *epistemic inheritance*: if the parent class has some interpretation, then all the things that are assigned to this class also inherit this interpretation.

Another class of epistemic conventions concerns *descriptions*, in which concepts are composed into sentences, such as "apples *grow on* trees," "apples *are* *larger than* grapes," and relationships between specific instances of concepts, like "this apple *is* rotten," "that grape *is smaller than* this apple," and so on. The semantic networks developed by psychologists and AI researchers in the 1960s and 1970s were the earliest knowledge representations of this type, but there have been many variants on this basic scheme. In the weakest description systems, there are no constraints; anything can enter into any kind of relationship with anything else. In stronger description systems, there will be more constraints on what is semantically meaningful. In the famous Chomskian sentence "colorless green ideas sleep furiously," the verb "sleep" is normally a relationship that constrains the noun phrase to refer to an object that is animate, not abstract. In a weak epistemic system, like a language that is only syntactically defined, this sentence is acceptable. In a stronger epistemic system that imposes semantic constraints on the objects that can enter into particular relationships or the relationships that can exist between certain kinds of objects, the sentence is unacceptable.

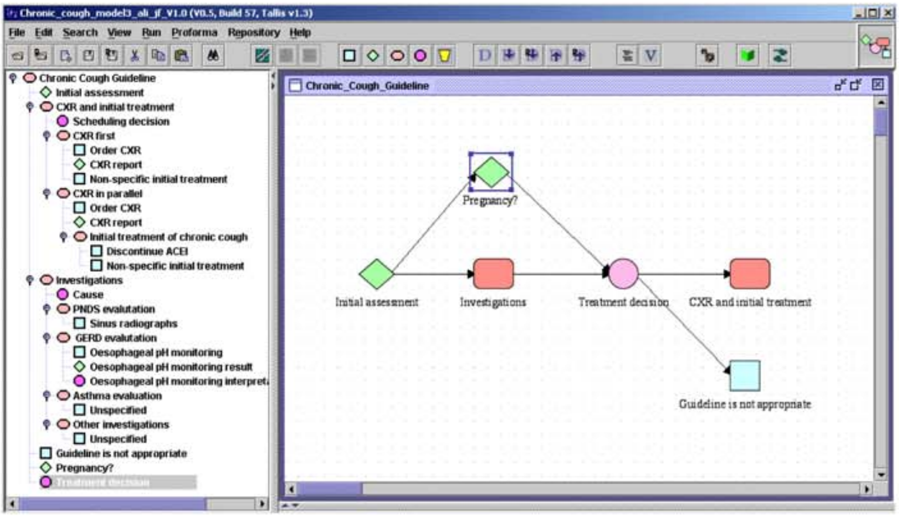
Rules can also be seen as a special kind of description. Logical implication, for example, can be viewed as a relationship between descriptions such that if one set of descriptions is true, then so is the other; from the logician's point of view, one set of descriptions has the role of "premises" and the other the status of "conclusions." If I say "Socrates is a man" and "all men are mortal," then you are entitled (under certain conventions about "all" that we agree about) to conclude that Socrates is mortal. Similarly, if I say that "Flora is an ancestor of Jane," and "Esther is an ancestor of Flora," then you are entitled under certain interpretations of the relationship *ancestor of* to conclude that "Esther is an ancestor of Jane." Some epistemic systems support higher-order properties such as transitivity of relations, so that if we assert that some relationship is transitive, as in *ancestor of*, then the knowledge system is sanctioned to deduce all valid ancestor relationships.

Since the introduction of the earliest knowledge representation systems, many different schemes have been developed that introduce further epistemic conventions and allow them to be combined. For example, "frame" systems exploit the epistemic convention of *ontological condensation*, in which collections of individual descriptions and rules that share a concept can be merged to represent a complex object. For example, an apple is a fruit that can be described as having a certain type of skin, color, and size, and if it is left too long, it will rot.

Attempts to build systems that can demonstrate high levels of cognitive functioning in complex domains are leading to more complex and abstract objects, or models. Two important types of model are *scenarios* and *tasks*. Scenarios are significant, distinctive situations that occur repeatedly (for example, in the medical domain, common scenarios are "patients with meningitis" or "being late for clinic"). Tasks are preplanned procedures that are appropriate in particular situations (such as "finding out what is wrong with someone with a high temperature" or "calling a taxi"). Despite their apparent complexity, many models can be understood as compositions of concepts, descriptions, rules, and, recursively, other models.

Rule-based languages have provided the basis of many successful knowledge representation systems, but task-based systems appear to be more powerful for modelling high-level cognitive skills (Fox, 2003). We have found in medicine that much of the expertise of skilled clinicians can be modelled in terms of a small set of tasks—plans, actions, and decisions—that can be composed into complex goal-directed processes. Tasks appear to have a small set of basic properties, including *preconditions* (descriptions of situations that must be true for the task to be relevant), *postconditions* (descriptions of the effects of carrying out the task), *trigger conditions* (scenarios in which the task is to be invoked), and *scheduling constraints* that describe sequential and temporal aspects of the process (as in finding out what is wrong with a patient before deciding what the treatment should be). Task-based models are being extensively investigated in medical AI (Peleg et al., 2003), and practical tools for formalizing this kind of knowledge are emerging (Figure 6).

Figure 6. PROforma process modelling system (Fox & Das, 2000) that is used for modelling expertise in terms of an ontology of tasks (plans, decisions, actions). A PROforma model is a hierarchy of plans containing decisions, actions and sub-plans (left panel) configured as a network (right panel, arrows indicate scheduling constraints. Rectangles = plans; circles = decisions, squares = actions, diamonds = enquiries, or actions that return information. PROforma models are automatically processed into a description language, which can be "enacted" by a suitable interpreter (e.g. Sutton & Fox, 2003).)


Chronic_cough_model3_ali_jf_V1.0 (V0.5, Build 57, Tallis v1.3)
File Edit Search View Run Proforma Repository Help
Chronic Cough Guideline
Initial assessment
CXR and initial treatment
Scheduling decision
CXR first
Order CXR
CXR report
Non-specific initial treatment
CXR in parallel
Order CXR
CXR report
Initial treatment of chronic cough
Discontinue ACEI
Non-specific initial treatment
Investigations
Cause
PNDS evalutation
Sinus radiographs
GERD evalutation
Oesophageal pH monitoring
Oesophageal pH monitoring result
Oesophageal pH monitoring interpret
Asthma evaluation
Unspecified
Other investigations
Unspecified
Guideline is not appropriate
Pregnancy?
Treatment decision
Chronic_Cough_Guideline
Pregnancy?
Initial assessment
Investigations
Treatment decision
CXR and initial treatment
Guideline is not appropriate

In recent years, many knowledge representation systems, from semantic networks and frame systems to rule-based systems, like production systems and logic-programming languages, have come to be seen as members of a family of knowledge representation systems called Description Logics (Baader et al., 2003). The early ideas about knowledge representation developed by psychologists (e.g., Quillian, 1968) stimulated many experimental investigations of the adequacy of particular languages. These have, in turn, given way to more formal theories of knowledge representation. The work of Brachman and his colleagues (1979, 1984; Nardi & Brachman, 2003) has been central in this development and has led to a much deeper understanding of the requirements on effective knowledge representation systems and their computational and other properties.

This brief presentation of basic ideas in knowledge representation does not do justice to the huge amount of work done in the last three decades. Recent developments in nonclassical and higher-order logics, for example, may yield important insights into epistemic properties of minds, both natural and artificial. Though such "exotic" representation systems face significant technical challenges, they may offer enormous computational power for cognitive systems in the future. We are certainly approaching a time when we can reasonably claim that we "know what knowledge is," and that we can predict from basic principles the properties that different knowledge representation systems will have. Description logics will provide some of the theory that will be needed to understand, and design, different kinds of minds.

Pathics: Minds and Mental States

A fundamental problem in developing ... a theory of rational agency is to give an account of the relationships that exist between an agent's mental states. In particular, a complete agent theory would explain how an agent's mental states lead it to select and perform rational actions ... M. Wooldridge, Reasoning about rational agents*, 2000 (p. 9)*

From the earliest philosophy to the beginnings of modern experimental psychology, ideas about mind have been influenced by introspection and intuition about everyday experience. This traditionally took the form of attempts to create credible narratives of experience with some appropriate vocabulary (from the ancient concepts of *nepesh*, *ruach*, and *leb* to modern ones like *memory* and *attention*) and common sense ideas like "beliefs," "feelings," and "points of view." The folk psychology of mind continues to evolve semi-independently of scientific investigation of mental processes. A concern that introspection and intuition are unreliable sources of insight into the human mind has led to some unwillingness to carry out research into mental states in the second half of the 20th century, leaving the field to philosophers, journalists, and authors of fiction.

Philosophers have comprised the main group that has taken mental states seriously and investigated them from within the framework of cognitive science. Daniel Dennett has influentially argued that mental states have been neglected, and that serious psychology and AI must adopt an "intentional stance" if they are to understand cognitive processes.[7] The intentional stance "is the strategy of interpreting the behaviour of an entity (person, animal, artefact, whatever) by treating it *as if* it were a rational agent who governed its 'choice' of 'action' by a 'consideration' of its 'beliefs' and 'desires'" (Dennett, 1996). Dennett draws attention to the limitations of the "physical stance" that is implicit in the static and dynamic models of cognition discussed above.

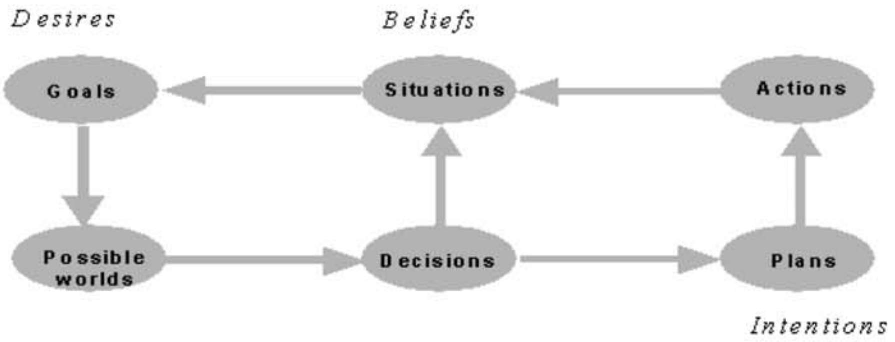
With the rise of AI and widespread discussions about whether machines can think, understand, etc., the anthropopathic[8] idea that machines might have mental states that resemble our own is now being pursued with some powerful tools, including cognitive modelling, mathematical logic, and formal epistemics. The philosophical idea of cognitive agents that possess "beliefs, desires, and intentions" has been particularly studied in software systems known as "BDI agents," where beliefs, desires, and intentions represent mental states that are being given clear (if speculative) semantics.

A number of systems have been developed that are claimed to be able to reason "rationally" over such states, e.g., Cohen and Levesque (1987), Rao and Georgeff (1995), Das et al. (1997), and Wooldridge (2000). Rao and Georgeff were the first to show that the BDI model could be used as a basis for building practical software agents. Fox and Das (2000) and Wooldridge (2000) discussed how the ideas can be captured in formal languages for specifying AI software of this kind.

BDI concepts can play an important role in understanding expert cognition in complex fields like medicine. Figure 7 illustrates an agent model that has emerged from my group's work on decision making and planning (Das et al., 1997; Fox & Das, 2000). Clinical expertise demands considerable cognitive flexibility, requiring the ability to reason over large collections of beliefs and knowledge, deal with high levels of uncertainty, and respond to unexpected circumstances (Fox, 2003). The domino model of mental processes (Figure 7) was designed to meet these requirements.

To illustrate how the model works, consider an imaginary medical situation, such as a patient presenting with an abnormal condition. Like any mental process, decision making and planning begin with some collection of *beliefs*, such as beliefs about a patient's state. Under certain circumstances, such as a dangerous clinical condition or trend, a clinician will adopt a *goal (*or *desire* in BDI terms)

Figure 7. BDI and Domino agent models



to eradicate, prevent, or preempt any disease that may be associated with the condition, first to determine what is wrong with the patient and then to determine the best treatment. Alternative possible diagnoses and treatments represent different *possible worlds* that the clinician must choose between. When there are a number of options in any situation, a *decision* must be made. Arguments for and against the different possible diagnoses and treatments for the patient can be developed and weighed in the balance (Fox & Parsons, 1998; Fox & Das, 2000). Depending upon the balance of argument, a particular *plan* (or *intention*) may be adopted and carried out in an appropriate schedule of *actions*. Actions may lead to new situations and beliefs, and the cognitive cycle continues.

The availability of a language of mental states could have important practical as well as theoretical uses. Without such concepts, we will have great difficulties understanding everyday human discourse, for example. Take the following remarks from a recent discussion of the professional responsibilities of a doctor (Pritchard, 2004):

.

- I am *committed* to putting the patient first.
- I practice *honestly, truthfully*, and with *integrity*.
- I respect *empathically* the patient's *identity, dignity, beliefs*, and *values*.
- I maintain confidentiality at all times.
- I am *open* to self-criticism and self-audit.
- I am *open* to peer criticism and peer audit.

- I *try* to provide care of high quality and apply evidence-based medicine where appropriate.
- I am *dedicated* to lifelong *reflection* and learning.

In these statements of professional principles, mentalistic terms (highlighted) are there in almost every line. The development of formal epistemics and pathics offers the possibility of making sense of such complex and subjective ideas. However, we should be cautious about what can presently be claimed. When we construct a formal BDI system, we are not defining a system with beliefs, desires, or intentions that are necessarily humanlike Yet, as Dennett has powerfully argued, if we ignore such aspects of intuition, we abandon a vast area of common sense phenomenology to "mere" philosophical and literary discussion. The formalization of BDI and similar agent systems is an interesting project in its own right and may provide us with a more precise way of talking about mental states than has ever previously been possible.

Discussion

The aim of this chapter has been to review a variety of images of mind, from the structure and operation of cognitive systems to ways of visualizing knowledge and mental states. We cannot understand the mind without being able to discuss the knowledge that makes it possible for it to interpret its environment, solve problems, and plan actions; the states it moves through in carrying out these tasks; and the control processes that make sure the whole shebang works properly to achieve its goals.

We need to bring together a number of types of accounts if we are to develop a general theory of mind. As in many sciences, from physics to biology, we need different kinds of theory to understand static structures from those that will shed light on dynamic behaviour. But the cognitive sciences need more. Cognitive agents have to be able to reflect on their knowledge and beliefs, make decisions and schedule actions in response, and anticipate possible consequences of their actions and plan for contingencies. For this, we need at least two types of theory that we do not need in the physical and biological sciences; *epistemic* theories are needed to understand the nature of knowledge and what makes a good representation, and *pathic* theories are needed to understand what Dennett would call the intentional aspects of cognition.

Even when we have reasonably complete accounts of all these levels, there will still be much to do. Many psychologists who are interested in explaining the foundations of mental states will, in the end, wish to ground their theories in neurology and biology. Broadbent made comments to that effect in 1958. With the rise of cognitive neuroscience, the time may be coming when we can make a reasonable fist of mapping down from an understanding of the functional architecture of the mind to the structural architecture of the brain. For example, the works of John Anderson[10] and Marcel Just, Pat Carpenter, and others[11] are showing promising progress on how the kinds of production rule systems that Newell introduced may be implemented in a way that is consistent with what we know about the static organization and dynamic operation of the human brain as revealed by physical imaging techniques, like positron-emission tomography and functional magnetic resonance.

What we would really like, of course, is a single theory that encompasses all these levels of description in one unified framework. That was what Newell was after. It is probably still asking too much, but one attempt along these lines may point us in a promising direction. This is a piece of work by David Glasspool, reported in another chapter in this volume. Glasspool's interests are in the convergence of theoretical ideas from cognitive neuroscience and AI, particularly the requirements that natural agents and artificial agents must both satisfy if they are to deal successfully with complex and unpredictable environments. His chapter explores a number of similarities between ideas in neuroscience and in agent research. He notes, for example, that natural cognitive processes and agent systems both require a combination of reactive control, to respond to unpredictable events in their environments, and deliberative control, to ensure continuity of purpose and behaviour over time.

Glasspool's model of high-level cognition is based on Shallice's (1988) account of human frontal lobe function and uses established concepts from cognitive- and neuropsychology brought together in a variant of the domino model. The static architecture of this part of the human cognitive system is modelled as a COGENT box and arrow diagram. Working memory for "beliefs" and another temporary store for plans ("schemas") are modelled as short-term storage buffers in COGENT, while dynamic aspects of the system are implemented by several distinct processing components. Low-level reactive operations are simulated using production rules, and high-level deliberative control is based on transitions between decisions and plan states, under the influence of belief and goal states stored in appropriate memory systems. An intermediate process that controls the selection of actions and plans implements a process called contention scheduling (Norman & Shallice, 1986; Shallice, 2002). Glasspool's demonstration suggests that a theory of cognitive functioning that unifies several very different views of mind may be within reach.

Conclusion

Boxes and arrows, ladders, dominos, and the other images of mind discussed in this chapter are just a few of many possible ways of visualizing cognition. The future will no doubt bring more. These images give important insights into the nature of cognition, but they are incomplete. A general theory of mind should unify the different perspectives. That task, however, demands more than images and metaphors. Only if we can translate our partial theories into a common, formal framework will we be able to achieve a unified theory that is consistent with the traditional physical account, while explaining intentional concepts like belief, goal, and commitment, and perhaps even ruach, leb, and nepesh.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought.* Mahwah, NJ: Erlbaum.

Baader, F., Valvanese, D., McGuinness, D., Nardi, D., & Patel-Schneider, P. (eds.). (2003). *The description logic handbook: Theory, implementation and applications*. London; New York: Cambridge: Cambridge University Press.

Brachman, R. J. (1979). On the epistemological status of semantic networks. In N. V. Findler (Ed.), *Associative networks* (pp. 3–50). New York: Academic Press.

Brachman, R. J., & Levesque, H. J. (1984). The tractability of subsumption in frame-based description languages. In *Proceedings of the Fourth National Conference on Artificial Intelligence* (AAAI84) (pp. 34–37).

Broadbent, D. E. (1958). *Perception and communication*. Oxford; Elmsford, NY: Pergamon Press.

Broadbent, D. E. (1971). *Decision and stress*. New York: Academic Press.

Chambers, W. & R. (Eds.). (1884). *Chambers' information for the people* (5th ed.). London: W&R Chambers.

Cohen, P. R., & Levesque, H. J. (1987). Persistence, intention and commitment. In M. P. Georgeff, & A. L. Lansky (Eds.), *Proceedings of the 1986 Workshop on Reasoning about Actions and Plans* (pp. 297–340). San Mateo, CA: Morgan Kaufmann.

Cooper, R., & Fox, J. (1998). COGENT: A visual design environment for cognitive modeling. *Behavior Research Methods, Instruments and Computers, 30*(4), 553–564.

Cooper, R., & Shallice, T. (1995). SOAR and the case for unified theories of cognition. *Cognition, 55*, 115–149.

Cooper, R., Fox, J., Farringdon, J., & Shallice, T. (1996). A systematic methodology for cognitive modeling. *Artificial Intelligence, 85*, 3–44.

Cooper, R. P. (2002). *Modeling high-level cognitive processes.* Mahwah, NJ: Lawrence Erlbaum.

Das, S., Fox, J, Elsdon, D., & Hammond, P. (1997). A flexible architecture for a general intelligent agent. *Journal of Experimental and Theoretical Artificial Intelligence, 9*, 407–440.

Dennett, D. C. (1996). *Kinds of minds: Towards an understanding of consciousness*. London: Weidenfeld and Nicholson.

Forgy, C., & McDermott, J. (1977). OPS, a domain independent production system language. *Proceedings of the Fifth International Joint Conference on Artifical Intelligence* (pp. 933–939), Cambridge, MA.

Fox, J. (2003). Logic, probability and the cognitive foundations of rational belief. *Journal of Applied Logic, 1*, 197–224.

Fox, J., Beveridge, M., & Glasspool, D. (2003). Understanding intelligent agents: Analysis and synthesis. *Artificial Intelligence Communications, 16*(3).

Fox, J., & Das, S. (2000). *Safe and sound: Artificial intelligence in hazardous applications.* Cambridge, MA: AAAI and MIT Press.

Fox, J., & Parsons, S. (1998). Arguing about beliefs and actions. In A. Hunter & S. Parsons (Eds.), *Applications of uncertainty formalisms*. Lecture Notes in Computer Science, 1455. Heidelberg: Springer.

Hajnal, S. J., Fox, J., & Krause, P. J. (1989). Sceptic User Manual. Available at *http://www.psychol.ucl.ac.uk/research/adrem/sceptic/manual_contents.html*

Just, M. A., Carpenter, P. A., & Varma, S. (1999). Computational modeling of high-level cognition and brain function. *Human Brain Mapping, 8*, 128–136.

Kowalski, R. (1979). *Logic for problem solving*. New York: Elsevier, North Holland.

Laird, J., Newell, A., & Rosenbloom, P. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*, 1–64.

MacDonald, P. (2003). *History of the concept of mind: Speculations about soul, mind and spirit from Homer to Hume.* Aldershot: Ashgate Publishing.

Nardi, D., & Brachman, R. (2003). An introduction to description logics. In F. Baader, D. Valvanese, D. McGuinness, D. Nardi, & P. Patel-Schneider (Eds.), *The description logic handbook*. London; New York: Cambridge University Press.

Neisser, U. (1967). *Cognitive psychology*. New York: Appleton Century Crofts.

Newell, A. (1973). Production systems: Models of control structures. In W. G. Chase (Ed.), *Visual information processing* (pp. 463–526). New York: Academic Press.

Newell, A. (1982). The knowledge level. *Artificial Intelligence, 18*(1), 87–127.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Newell, A., & Simon, H. A. (1972). *Human problem solving*. Englewood Cliffs, NJ: Prentice Hall.

Norman, D., & Shallice, T. (1986). Attention to action: Willed and automatic control of behavior. In R. Davidson, G. Schwartz, & D. Shapiro (Eds.), *Consciousness and self regulation: Advances in research and theory* (Vol. 4; pp. 1–18). News York: Plenum.

Peleg, M., Tu, S., Bury, J., Ciccarese, P., Fox, J., Greenes, R. A., Hall, R., Johnson, P. D., Jones, N., Kumar, A., Miksch, S., Quaglini, S., Seyfang, A., Shortliffe, E. H., & Stefanelli, M. (2003). Comparing computer-interpretable guideline models: A case-study approach. *Journal of the American Medical Informatics Association, 10*(1, Jan.–Feb.), 52–68.

Pritchard, P. (To appear). Professional values and informatics: What is the connection?

Quillian, M. (1968). Semantic memory. In M. Minsky (Ed.), *Semantic information processing*. Cambridge, MA: MIT Press.

Rao, A. S., & Georgeff, M. P. (1995). BDI agents: From theory to practice. *Proceedings of the First International Conference on Multi-Agent System*s (ICMAS-95), San Francisco, CA.

Shallice, T. (2002). Fractionation of the supervisory system. In D. T. Stuss & R. Knight (Eds.), *Principles of frontal lobe function* (pp. 261–277). New York: Oxford.

Solso, R. L. (1998). *Cognitive psychology* (5th ed.). Needham Heights: Allyn and Bacon.

Sutton, D., & Fox, J. (2003). The syntax and semantics of the PROforma guideline modeling language. *Journal of the American Medical Informatics Association, 10*, 433–443.

Wooldridge, M. (2000). *Reasoning about rational agents*. Cambridge, MA: MIT Press.

Endnotes

[1] Paul MacDonald, personal communication.

[2] For more on the early history of AI, see http://livinginternet.com/i/ii_ai.htm

[3] There are reasons to consider this simplistic today, as we shall see in the next section, but it is a good scientific strategy to work with a simple hypothesis until there is a compelling reason to complicate things.

[4] COGENT can be downloaded from http://cogent.psyc.bbk.ac.uk/

[5] This pedagogical device cannot do justice to the great body of research in this field, but, as with the rest of this discussion, my purpose is not to offer a thorough review of the subject but to give a flavor of what is going on.

[6] Most of the work on formal representation is being carried out in AI rather than psychology, and I have been asked whether this implies a rejection of connectionist or "subsymbolic" theories of knowledge, which are more influential in psychology and neuroscience. I think it does not. Whether one believes the human brain-mind system is "really" symbolic or not, we must use symbolic formalisms if we are to develop theory in this field. We describe the acceleration of an object falling in a gravitational field in a symbolic mathematical language but do not conclude from this that the object is carrying out some sort of symbolic computation.

[7] For more on Dennett's perspective, see http://ase.tufts.edu/cogstud/~ddennett.htm

[8] ascribing human feelings to a god or inanimate object"

[9] For this reason, I prefer not to make an anthropopathic attribution to BDI agents and prefer simply to refer to this kind of theory as "pathic" (as in empathic, telepathic, and psychopathic).

[10] For more on Anderson's ACT-R, see http://act-r.psy.cmu.edu/

[11] See www.ccbi.cmu.edu/ for more on 4CAPS and the work of Just and Carpenter.

Chapter 8

A "Consciousness"-Based Architecture for a Functioning Mind

Stan Franklin
The University of Memphis, USA

Abstract

Here we describe an architecture for an autonomous software agent designed to model a broad spectrum of human cognitive and affective functioning. In addition to featuring "consciousness," the architecture accommodates perception, several forms of memory, emotions, action-selection, deliberation, ersatz language generation, several forms of learning, and metacognition. One such software agent, IDA, embodying much of this architecture, is up and running. IDA's "consciousness" module is based on global workspace theory, allowing it to select relevant resources with which to deal flexibly with both exogenous and endogenous stimuli. Within this architecture, emotions implement IDA's drives, its[1] *primary motivations. Offering one possible architecture for a fully functioning artificial mind, IDA constitutes an early attempt at the exploration of design space and niche space. The design of the IDA architecture spawns hypotheses concerning human cognition and affect that can serve to guide the research of cognitive scientists and neuroscientists. One such hypothesis is that consciousness is discrete.*

Introduction

What is a mind? I have maintained for years, and still do, that the most useful way to look at a mind is as a control structure for an autonomous agent (see the next section). The continuing task of a mind is to produce the agent's next action, to answer the only really significant question there is—what shall I do next (Franklin, 1995). Any theory specifying how to go about answering this question is a theory of mind.[2] A theory is computationally plausible if it can be implemented or modeled on a computer, very likely on a virtual machine running on a computer (Sloman & Chrisley, 2003). If our theory is to be implemented or modeled on a computer, we must have in hand a computationally plausible architecture with which to implement or model it. If we have succeeded in implementing our architecture on a computer so that it supports our theory of mind on an autonomous agent, we have produced an artificial mind.

This chapter is devoted primarily to the description of one such, complex, functioning, artificial mind, and to some of the hypotheses about human affect and cognition that are derived from it. This artificial mind is the control structure of an autonomous software agent, IDA (Franklin, Kelemen, & McCauley, 1998; Franklin, 2001). IDA's architecture implements global workspace theory, a theory of mind (Baars, 1988, 1997, 2002). It can be seen as an early contribution to the exploration of design space and niche space (Sloman, 1998).

Autonomous Agents

Artificial intelligence pursues the twin goals of understanding human intelligence and of producing intelligent software and artifacts. Designing, implementing, and experimenting with autonomous agents furthers both of these goals in a synergistic way (Franklin, 1997). An *autonomous agent* (Franklin & Graesser, 1997) is a system situated in, and part of, an environment, which senses that environment, and acts on it, over time, in pursuit of its own agenda. In biological agents, this agenda arises from evolved drives and their associated goals; in artificial agents from drives and goals built in by its designer. Such drives that act as motive generators (Sloman, 1987) must be present, whether explicitly represented or expressed causally. The agent also acts in such a way as to possibly influence what it senses at a later time. In other words, it is structurally coupled to its environment (Maturana, 1975; Maturana et al., 1980). Biological examples of autonomous agents include humans and most animals. Nonbiological examples include some mobile robots and various computational agents, including artificial life agents, software agents, and many computer viruses. We will be concerned with autonomous software agents designed for specific tasks and "living" in real-world computing systems, such as operating systems, databases, or networks.

Such autonomous software agents serve to spawn hypotheses about human cognition and affect that can serve to guide the research of cognitive scientists and neuroscientists. Each design decision taken for the agent translates into such a hypothesis about human cognitive or affective functioning (Franklin, 1997). Thus, in addition to their practical function, such agents can further the interests of science.

Roboticists often claim that autonomous software agents are not embodied in that they typically do not have to deal with the physics of the real world (Prem, 1997). However, some software agents, including our IDA, negotiate with humans in a real-world environment. They both causally affect the real, physical world and are affected by it. For this to happen, they, in some sense, must be embodied.

Global Workspace Theory

The material in this section relates to Baars' two books (1988, 1997) and superficially describes his global workspace theory of consciousness. In this theory, Baars, along with many others (for example, Minsky, 1985; Ornstein, 1986; Edelman, 1987; Jackson, 1987), postulates that human cognition is largely implemented by a multitude of relatively small, special-purpose processes, almost always subconscious. (It is a multiagent system.) Communication between them is rare and over a narrow bandwidth. Coalitions of such processes find their way into a global workspace (and thus into consciousness). This limited capacity workspace serves to broadcast the message (contents) of the coalition to all the subconscious processors in order to recruit other processors to join in the handling of the current novel situation or in solving the current problem. Thus, consciousness in this theory allows us to deal with novelty or problematic situations that cannot be dealt with efficiently, or at all, by automatized, subconscious processes. In particular, consciousness provides access to appropriately useful resources, thereby solving the *relevance problem*, that is, the problem of identifying those resources that are relevant to the current situation.

All of this takes place under the auspices of contexts: goal contexts, perceptual contexts, conceptual contexts, and cultural contexts. Baars uses goal hierarchies, dominant goal contexts, a dominant goal hierarchy, dominant context hierarchies, and lower level context hierarchies. Each context is a coalition of processes. Though contexts are typically subconscious, they strongly influence conscious processes.

Baars postulated that learning results simply from conscious attention; that is, that consciousness is sufficient for learning. There is much more to the theory, including attention, action selection, emotion, voluntary action, metacognition, and a sense of self. I think of it as a high-level theory of cognition and of affect.

"Conscious" Software Agents

A *"conscious"* software agent is defined to be an autonomous software agent that implements global workspace theory. (No claim of sentience or phenomenal consciousness is being made, hence, the scare quotes.) I believe that "conscious" software agents have the potential to play a synergistic role in both cognitive theory and intelligent software. Minds can be viewed as control structures for autonomous agents (Franklin, 1995). A theory of mind constrains the design of a "conscious" agent that implements that theory. While a theory is typically abstract and only broadly sketches an architecture, an implemented, computational design provides a fully articulated architecture and a complete set of mechanisms. This architecture and set of mechanisms provides a richer, more concrete, and more decisive theory. Moreover, every design decision taken during an implementation furnishes a hypothesis about how human minds work. These hypotheses may motivate experiments with humans and other forms of empirical tests, thereby providing direction to research in cognitive science and neuroscience. Conversely, the results of such experiments motivate corresponding modifications of the architecture and mechanisms of the software agent. In this way, the concepts and methodologies of cognitive science and of computer science will work synergistically to enhance our understanding of mechanisms of mind (Franklin, 1997).

"Conscious" Mattie

"Conscious" Mattie (CMattie) was our first "conscious" software agent. She is a clerical agent devoted to publicizing mostly weekly seminars on various subjects in a Mathematical Sciences Department (McCauley & Franklin, 1998; Ramamurthy et al., 1998; Zhang et al., 1998; Bogner et al., 2000). She composes and e-mails weekly seminar announcements, having communicated by e-mail with seminar organizers and announcement recipients in natural language. She maintains her mailing list, reminds organizers who are late with their information, and warns of space and time conflicts. There is no human involvement other than these e-mail messages. CMattie's cognitive modules include perception, learning, action selection, associative memory, "consciousness," emotion, and metacognition. Her emotions influence her action selection. Her mechanisms include variants and extensions of Maes' behavior nets (1989), Hofstadter and Mitchell's Copycat architecture (1994), Jackson's pandemonium theory (1987), Kanerva's sparse distributed memory (1988), and Holland's classifier systems (Holland, 1986).

CMattie will play only a minor role in what follows. The brief description above is included for two reasons. Several of the references given in the context of IDA's modules will, in fact, describe similar modules in CMattie rather than IDA modules. In these cases, the descriptions therein will be mostly accurate when applied to the corresponding IDA module. Second, CMattie constitutes an instructive example relating to the exploration of design space (Sloman, 1998). CMattie's cognitive processes are reactive and metacognitive without being deliberative, demonstrating that this combination is possible if only in an impoverished way.

IDA

IDA (Intelligent Distribution Agent) is a "conscious" software agent that was developed for the U.S. Navy (Franklin et al., 1998). At the end of each sailor's tour of duty, he or she is assigned to a new billet. This assignment process is called distribution. The Navy employs some 280 people, called detailers, full time, to effect these new assignments. IDA's task is to facilitate this process by automating the role of detailer.

IDA's task presents both communication problems and action selection problems involving constraint satisfaction. She must communicate with sailors via e-mail and in natural language, understanding the content and producing lifelike responses. Sometimes IDA will initiate conversations. She must access a number of databases, again understanding the content. She must see that the Navy's needs are satisfied, for example, the required number of sonar technicians on a destroyer with the required types of training. In doing so, IDA must adhere to a number of Navy policies. She must hold down moving costs. And, IDA must cater to the needs and desires of the sailors as well as is possible. This includes negotiating with the sailor via an e-mail correspondence in natural language. IDA's architecture and mechanisms are largely modeled after those of CMattie, though they are more complex. In particular, IDA employs deliberative reasoning in the service of action selection, where CMattie was able to do without.

Before going further, it is important that we distinguish between IDA as a computational model and as a conceptual model. The computational IDA is a running piece of Java code, an actual software agent. The conceptual IDA model includes everything in the computational model with relatively minor changes. It also includes, however, additional functionality that has been designed but not yet implemented. In what follows, I will try to carefully note capabilities not yet implemented. Unless otherwise stated, descriptions will be of the computational model.

As we hypothesize that humans also do, the IDA model runs in a rapidly continuing sequence of partially overlapping cycles, called cognitive cycles (Baars & Franklin, 2003). These cycles will be discussed in detail below, after the IDA architecture and its mechanisms are described.

"Conscious" Software Architecture and Mechanisms

The IDA architecture is partly symbolic and partly connectionist, at least in spirit. Although there are no artificial neural networks as such, spreading activation abounds. The mechanisms used in implementing the several modules have been inspired by a number of different new AI techniques (Hofstadter & Mitchell, 1994; Holland, 1986; Jackson, 1987; Kanerva, 1988; Maes, 1989; Minsky, 1985; Valenzuela-Rendon, 1991). The architecture is partly composed of entities at a relatively high level of abstraction, such as behaviors, message-type nodes, emotions, etc. (all discussed in this chapter), and partly of low-level codelets.

Each codelet is a small piece of code performing a simple, specialized task. They correspond to Baars' processors in global workspace theory (1988). Most codelets are, like demons in an operating system, always watching for a situation to arise, making it appropriate to act. Codelets come in many varieties: perceptual codelets, information codelets, attention codelets, behavior codelets, expectation codelets, etc. (all described in this chapter). Though most codelets subserve some high-level entity, many codelets work independently. Codelets do almost all the work. IDA can almost be viewed as a multiagent system, though not in the usual sense of the term.

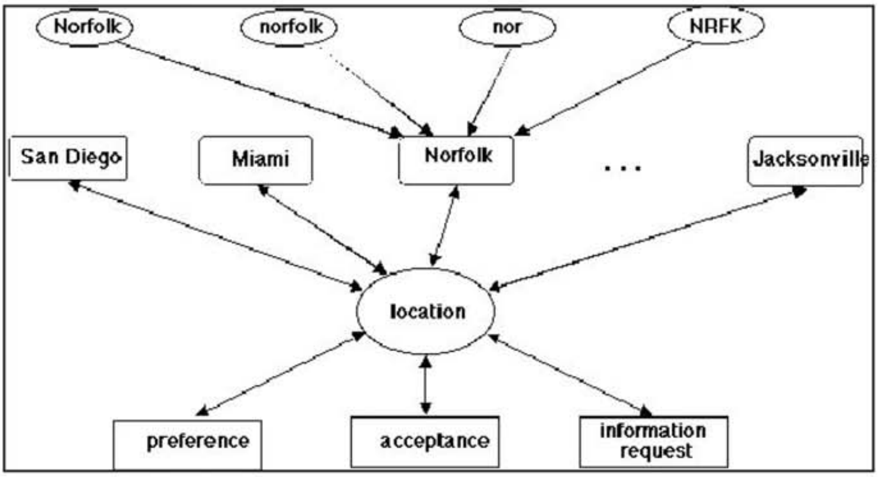
As noted above, most of IDA's various entities, both high-level entities and codelets, carry and spread some activation. Such activation typically hopes to measure some sort of strength or relevance. Unless told otherwise, it is safe to assume that every activation decays over time. Finally, note that though the IDA architecture is conveniently described in terms of modules, it is, in fact, tightly connected. Like the brain, the IDA architecture is both modular and highly interconnected.

Perception

IDA perceives both exogenously and endogenously (Zhang et al., 1998). The stimuli of her single sense are strings of characters. We use Barsalou's perceptual symbol systems as a guide (1999). The perceptual knowledge base of this agent, called perceptual memory, takes the form of a semantic net with activation called the slipnet. The name is taken from the Copycat architecture that employs a similar construct (Hofstadter & Mitchell, 1994). Nodes of the slipnet constitute the agent's perceptual symbols, representing individuals, categories, and higher-level ideas and concepts. A link of the slipnet represents a relation between its source node and its sink node (see Figure 1).

An incoming stimulus, say an e-mail message, is descended upon by a hoard of perceptual codelets. Each of these codelets is looking for some particular string or strings of characters, say one of the various forms of the name of the city of Norfolk. Upon finding an appropriate character string, the codelet will activate an appropriate node or node in the slipnet. The slipnet will eventually settle down. Nodes with activations over threshold and their links are taken to be the constructed meaning of the stimulus. Pieces of the slipnet containing nodes and links, together with perceptual codelets with the task of copying the piece to working memory constitute Barsalou's perceptual symbol simulators.

Figure 1. A portion of the Slipnet



Memory

Both CMattie and IDA employ sparse distributed memory (SDM) as their major associative memories (Anwar, Dasgupta, & Franklin, 1999; Anwar & Franklin, 2003). SDM is a content addressable memory that, in many ways, is an ideal computational mechanism for use as a long-term associative memory (Kanerva, 1988). Content addressable means that items in memory can be retrieved by using part of their contents as a cue, rather than having to know the item's address in memory.

The inner workings of SDM rely on large binary spaces, that is, spaces of vectors containing only zeros and ones, called bits. These binary vectors, called words, serve as both the addresses and the contents of the memory. The dimension of the space determines the richness of each word. These spaces are typically far too large to implement in any conceivable computer. Approximating the space uniformly with some manageable number of actually implemented, hard locations surmounts this difficulty. The number of such hard locations determines the carrying capacity of the memory. Features are represented as one or more bits. Groups of features are concatenated to form a word. When writing a word to memory, a copy of the word is placed in all close enough hard locations. When reading a word, a close enough cue would reach all close enough hard locations and get some sort of aggregate or average out of them. As mentioned above, reading is not always successful. Depending on the cue and the previously written information, among other factors, convergence or divergence during a reading operation may occur. If convergence occurs, the pooled word will be the closest match (with abstraction) of the input reading cue. On the other hand, when divergence occurs, there is no relation, in general, between the input cue and what is retrieved from memory.

SDM is much like human long-term declarative memory. A human often knows what he or she does or does not know. If asked for a telephone number I have once known, I may search for it. When asked for one I have never known, an immediate "I don't know" response ensues. SDM makes such decisions based on the speed of initial convergence. The reading of memory in SDM is an iterative process. The cue is used as an address. The content at that address is read as a second cue, and so on, until convergence, that is, until subsequent contents look alike. If it does not quickly converge, an "I don't know" is the response. The "on the tip of my tongue phenomenon" corresponds to the cue having content just at the threshold of convergence. Yet another similarity is the power of rehearsal, during which an item would be written many times and, at each of these, to a thousand locations—that is the *distributed* part of sparse distributed memory. A well-rehearsed item can be retrieved with smaller cues. Another similarity is interference, which would tend to increase over time as a result of other similar writes to memory. The IDA conceptual model uses variants of SDM to implement both transient episodic memory and declarative memory (Franklin et al., in review; Ramamurthy, D'Mello, & Franklin, to appear).

"Consciousness"

The "consciousness" modules in CMattie and IDA are almost identical. In both architectures, the processors postulated by global workspace theory are implemented by codelets, small pieces of code. These are specialized for some simple task and often play the role of a demon waiting for an appropriate condition under which to act. The apparatus for producing "consciousness" consists of a coalition manager, a spotlight controller, a broadcast manager, and a collection of attention codelets that recognize novel or problematic situations (Bogner, 1999; Bogner et al., 2000). Each attention codelet keeps a watchful eye out for some particular situation to occur that might call for "conscious" intervention. Upon encountering such a situation, the appropriate attention codelet will be associated with the small number of information codelets that carry the information describing the situation. This association should lead to the collection of this small number of codelets, together with the attention codelet that collected them, becoming a coalition. Codelets also have activations. Upon noting a suitable situation, an attention codelet will increase its activation as a function of the match between the situation and its preferences. This allows the coalition, if one is formed, to compete for "consciousness."

The coalition manager is responsible for forming and tracking coalitions of codelets. Such coalitions are initiated on the basis of the mutual associations between the member codelets. During any given cognitive cycle, one of these coalitions finds its way to "consciousness," chosen by the spotlight controller, who picks the coalition with the highest average activation among its member codelets. Global workspace theory calls for the contents of "consciousness" to be broadcast to each of the codelets. The broadcast manager accomplishes this.

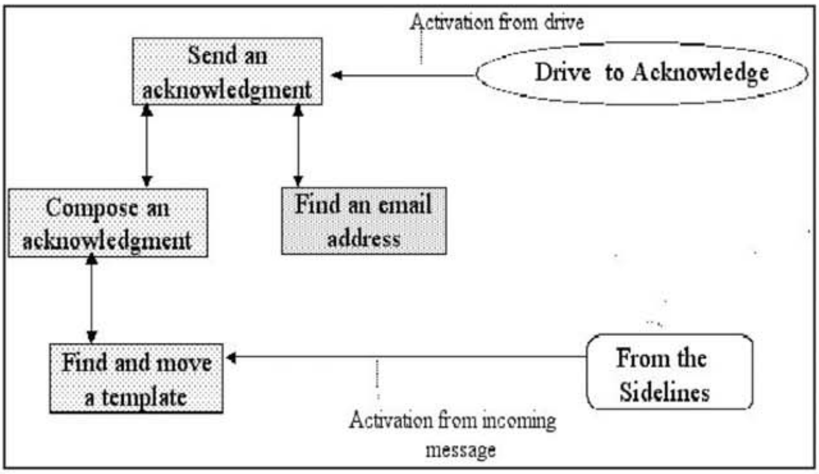
Action Selection

CMattie and IDA depend on an enhancement of Maes' behavior net (1989) for high-level action selection in the service of built-in drives (Song & Franklin, 2000; Negatu & Franklin, 2002). Each has several distinct drives operating in parallel and implemented in the IDA conceptual model by feelings and emotions. These drives vary in urgency as time passes and the environment changes. The goal contexts of global workspace theory are implemented as *behaviors* in the IDA model. Behaviors are typically mid-level actions, many depending on several behavior codelets for their execution. A behavior net is composed of behaviors and their various links. A behavior looks very much like a production rule, having preconditions as well as additions and deletions. A behavior is distinguished from a production rule by the presence of an activation, which is a number indicating some kind of strength level. Each behavior occupies a node in a digraph (directed graph). The three types of links of the digraph are completely determined by the behaviors. If a behavior *X* will add a proposition *b*, which is on behavior *Y*'s precondition list, then put a successor link from *X* to *Y*. There may be several such propositions, resulting in several links between the same nodes. Next, whenever you put in a successor going one way, put in a predecessor link going the other. Finally, suppose you have a proposition *m* on behavior *Y*'s delete list that is also a precondition for behavior *X*. In such a case, draw a conflictor link from *X* to *Y*, which is to be inhibitory rather than excitatory.

As in connectionist models, this digraph spreads activation. The activation comes from four sources: from activation stored in the behaviors, from the environment, from drives (through feelings and emotions in the IDA conceptual model), and from internal states. The environment awards activation to a behavior for each of its true preconditions. The more relevant it is to the current situation, the more activation it is going to receive from the environment. This source of activation tends to make the system opportunistic. Each drive awards activation to every behavior that, by being active, will satisfy that drive. This source of activation tends to make the system goal directed. Certain internal states of the agent can also send activation to the behavior net. This activation, for example, might come from a coalition of behavior codelets responding to a "conscious" broadcast. Finally, activation spreads from behavior to behavior along links. Along successor links, one behavior strengthens those behaviors with preconditions that it can help fulfill by sending them activation. Along predecessor links, one behavior strengthens any other behavior with an add list that fulfills one of its own preconditions. A behavior sends inhibition along a conflictor link to any other behavior that can delete one of its true preconditions, thereby weakening it. Every conflictor link is inhibitory. A behavior is *executable* if all of its preconditions are satisfied. To be acted upon, a behavior must be executable, must have activation over threshold, and must have the highest such activation. Behavior nets produce flexible, tunable action selection for these agents.

Behaviors in these agents almost always operate as part of behavior streams, which correspond to goal context hierarchies in global workspace theory. Visualize a behavior stream as a subgraph of the behavior net, with its nodes connected by predecessor links (Figure 2). A behavior stream is sometimes a sequence, but not always. It can fork in either a forward or backward direction. A behavior stream can be usefully thought of as a partial plan of action.

Figure 2. Behavior stream



Constraint Satisfaction

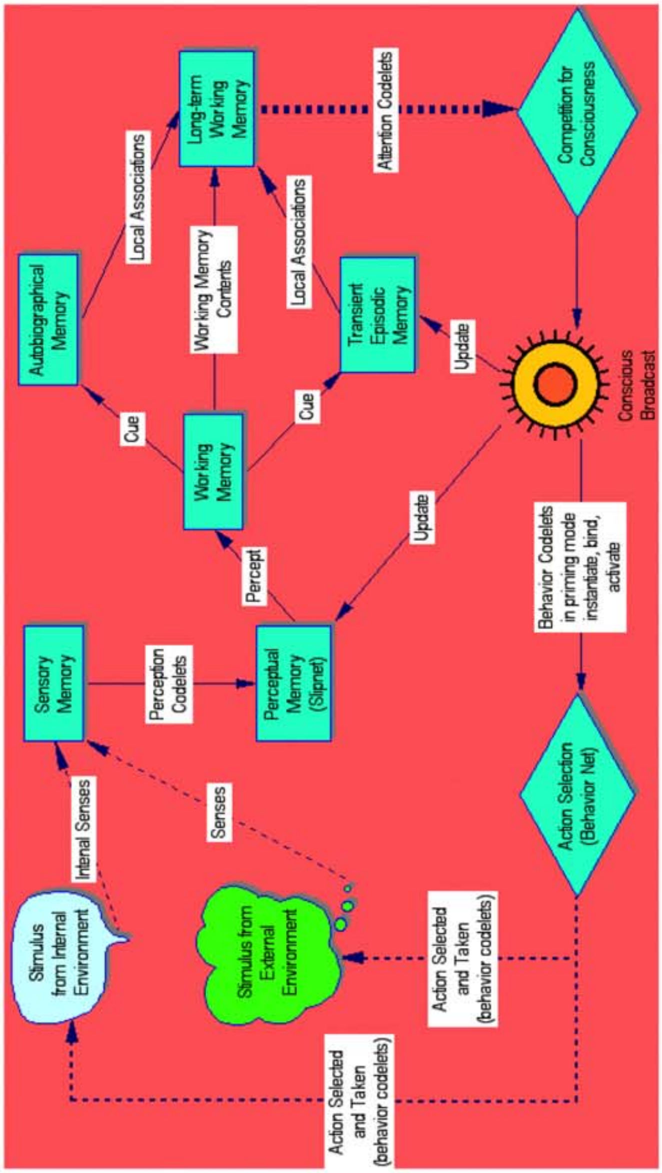
At the heart of IDA's task of finding new jobs for sailors lies the issue of constraint satisfaction. Not only must IDA consider the preferences of the sailor, she must also see that the requirements of an individual job are met, and simultaneously adhere to the policies of the Navy. Taking such issues into consideration, IDA's constraint satisfaction module is designed to provide a numerical measure of the fitness for a particular job for a particular sailor.

Given a specified issue such as a sailor preference, a particular Navy policy, or specific job requirement, a function is defined that provides a numerical measure of the fitness of this job for this sailor with respect to this particular issue. Computationally, these functions are diverse and often nonlinear. Most take their input from information from the sailor's personnel record or from the job requisition list that has already been written to IDA's working memory.

To find a common currency for these various issues, we laboriously found a set of weights for the issues, measuring their relative importance (Keleman et al., 2002). With these weights in hand, the weighted sum over the various issues forms a linear functional that provides the desired numerical measure of fitness. The calculating of such a fitness value requires several of IDA's cognitive cycles (see the following) to process each in the linear functional.

This is an oversimplified account; for example, there are multiplicative terms dealing with hard constraints. I chose not to describe IDA's constraint satisfaction module in more detail, because it is so restricted to her particular, practical domain. In some other application of the IDA architecture, the constraint satisfaction module would not likely appear.

Figure 3. IDA's cognitive cycle


Stimulus from Internal Environment
Stimulus from External Environment
Intenal Senses
Senses
Sensory Memory
Perception Codelets
Perceptual Memory (Slipnet)
Percept
Working Memory
Cue
Cue
Autobiographical Memory
Local Associations
Working Memory Contents
Local Associations
Transient Episodic Memory
Long-term Working Memory
Attention Codelets
Competition for Consciousness
Update
Update
Conscious Broadcast
Behavior Codelets in priming mode instantiate, bind, activate
Action Selection (Behavior Net)
Action Selected and Taken (behavior codelets)
Action Selected and Taken (behavior codelets)

Deliberation

Action selection via the behavior net suffices for CMattie due to her relatively constrained domain. IDA's domain is more complex and requires deliberation in the sense of creating possible scenarios and partial plans of actions and then choosing between them. For example, suppose IDA is considering a ranked list of several possible jobs for a given sailor produced by her constraint satisfaction module, all seemingly suitable. IDA must construct a scenario for at least one of these possible billets. In each scenario, the sailor leaves his or her current position during a certain time interval, spends a specified length of time on leave, possibly reports to a training facility on a certain date, and arrives at the new billet within a given time window. Such scenarios are judged on how well they fit the temporal constraints and on moving and training costs.

As in humans, deliberation is mediated by the "consciousness" mechanism. Imagine IDA in the context of a behavior stream with a goal to construct a scenario to help evaluate a particular job for a particular sailor. She must first decide on a departure date within an allowable window, the first event of the scenario. Then, events for travel time (often in more than one segment), leave time (occasionally in several segments), training time (with specified dates), and arrival date must be decided upon, again within an appropriate window. If the first try does not work, IDA typically starts over with a suitably adjusted departure date. If still unsuccessful after several tries, IDA will give up on that particular job and go on to another. When successful, the job in question is so marked in working memory and becomes a candidate for voluntary selection (see below) to be offered to the sailor. Each step in this process will require several cognitive cycles, as described below. Thus, IDA is capable of temporal deliberation.

Voluntary Action

Deliberation is also used in IDA to implement voluntary action in the form of William James' ideomotor theory as prescribed by global workspace theory (Baars, 1988, Chapter VII). Suppose scenarios have been constructed for several of the more suitable jobs. An attention codelet spots one that it likes, possibly due to this codelet's predilection for, say, low moving costs. The act of an attention codelet's bringing one of these candidates to consciousness serves to propose it. This is James' idea popping into mind. If no other attention codelet brings an objection to consciousness or proposes a different job, a timekeeper codelet assigned the particular task of deciding will conclude, after a suitable time having passed, that the proposed job will be offered, and starts the process by which it will be so marked in working memory. Objections and proposals can continue to come to consciousness, but the patience of the timekeeper codelet dampens as time passes. Also, the activation of a given attention codelet tends to diminish after winning a competition for consciousness in any given cognitive cycle. This lessening makes it less likely that this particular attention codelet will be successful in proposing, objecting, or supporting in the near future. This diminishing of patience and activation serve to prevent continuing oscillations in the voluntary action selection process.

Language Generation

IDA's language generation module follows the same back and forth to "consciousness" routine carried out over a number of cognitive cycles. For example, in composing a message offering a sailor a choice of two billets, an attention codelet would bring to "consciousness" the information that this type of message was to be composed and the sailor's name, pay grade, and job description. After the "conscious" broadcast and the involvement of the behavior net as described above, a script containing the salutation appropriate to a sailor of that pay grade and job description would be written to the working memory. Another attention codelet would bring this salutation to "consciousness," along with the number of jobs to be offered. The same process would result in an appropriate introductory script being written below the salutation. Continuing in this manner, filled in scripts describing the jobs would be written, and the message would be completed. Note that different jobs may require different scripts. The appeal to "consciousness" results in some version of a correct script being written.

The mediation by the "consciousness" mechanism, as described in the previous paragraphs, is characteristic of IDA. The principle is that IDA should use "consciousness" whenever a human detailer would be conscious in the same situation. For example, IDA could readily recover all the needed items from a sailor's personnel record subconsciously with a single behavior stream. But, a human detailer would be conscious of each item individually. Hence, according to our principle, so must IDA be "conscious" of each retrieved personnel data item.

This completes our description of the IDA computational model, which implements small parts of what is described below as part of the IDA conceptual model. We now have more than enough of the architecture in hand to discuss the cognitive cycle that derives from the IDA model.

IDA Cognitive Cycle

As noted above, the primary question facing the control system (mind) of any autonomous agent at any moment is, "What do I do now?" IDA answers this question in a moment-to-moment fashion by means of a continuing repetition of her cognitive cycle.[3] One can usefully decompose this cognitive cycle into the nine steps described just below. We choose to follow the computer science fashion (input, processing, output) and the psychological fashion (stimulus, cognition, response) by beginning the cycle with perception and ending it with an action. Note that many, if not most, of our actions aim at choosing the next sensory experience, suggesting that the cycle might well be regarded a beginning with an action. This is also true of some of IDA's actions, particularly her consultation of various Navy databases. Because we hypothesize that we humans employ just such a cognitive cycle (Franklin et al., In review), our description will include elements postulated for humans that are not present in IDA. These will be noted. The description will also mention as yet unimplemented capabilities from the IDA conceptual model (see the following). Here is the description of IDA's cognitive cycle (Baars & Franklin, 2003):

1. **Perception.** Sensory stimuli, external or internal, are received and interpreted by perception assigning meaning. Note that this stage is subconscious.
   a) *Early perception*. Input arrives through senses. Specialized perceptual codelets descend on the input. Those that find features relevant to their specialty activate appropriate nodes in the slipnet (Perceptual Memory), a semantic net with activation. (For example, a codelet finding the string Norfolk would activate a Norfolk node.)
   b) *Chunk perception*. Activation passes from node to node in the slipnet (for example, from Norfolk to Location). The slipnet stabilizes, bringing about the convergence of streams from different senses (in humans) and chunking bits of meaning into larger chunks. These larger chunks, represented by meaning nodes in the slipnet, constitute the percept. (For example, Preference for a Particular Location.)
2. **Percept to preconscious buffer.** The percept, including some of the data plus the meaning, is stored in preconscious buffers of IDA's working memory. In humans, these buffers may involve visuospatial, phonological, and other kinds of information. (For example, in the computational IDA, a percept might include a nine-digit number tagged as a social security number, or a text string tagged as a location, or the recognition of a stated preference for a particular location.)
3. **Local associations.** Using the incoming percept and the residual contents of the preconscious buffers as cues, local associations are automatically retrieved from transient episodic memory (Taylor, 1999; Conway, 2001; Donald, 2001, p. 137) and from long-term declarative memory. The contents of the preconscious buffers together with the retrieved local associations from transient episodic memory and long-term associative memory. Together, these roughly correspond to Ericsson and Kintsch's long-term working memory (1995) and Baddeley's episodic buffer (Baddeley, 1993).
4. **Competition for consciousness.** Attention codelets, with the job of bringing relevant, urgent, or insistent events to consciousness, view long-

term working memory. Some of them gather information, form coalitions, and actively compete for access to consciousness. (For example, in the computational IDA, an attention codelet may gather into a coalition an information codelet carrying the rate and name AS1 Kevin Adams, another carrying the location Norfolk, and yet another, the idea Preference for a Particular Location.) The competition may also include attention codelets from a recent previous cycle. The activation of unsuccessful attention codelets decays, making it more difficult for them to compete with newer arrivals. However, the contents of unsuccessful coalitions remain in the preconscious buffer and can serve to prime ambiguous future incoming percepts. The same is true of contents of long-term working memory that are not picked up by an attention codelet.

5. **Conscious broadcast.** A coalition of codelets, typically an attention codelet and its covey of related information codelets carrying content, gains access to the global workspace and has its contents broadcast. (For example, IDA may become "conscious" of AS1 Kevin Adams preference for being stationed in Norfolk.) This broadcast is hypothesized to correspond to phenomenal consciousness[4] in humans. The current contents of consciousness are also stored in transient episodic memory. At recurring times, not part of a cognitive cycle, the contents of transient episodic memory are consolidated into long-term associative memory (Shastri, 2001, 2002). Transient episodic memory is an associative memory with a relatively fast decay rate (Conway, 2001). It is to be distinguished from autobiographical memory, a part of long-term declarative memory.

6. **Recruitment of resources.** Relevant behavior codelets respond to the conscious broadcast. These are typically codelets with variables that can be bound from information in the conscious broadcast. If the successful attention codelet was an expectation codelet calling attention to an unexpected result from a previous action, the responding codelets may be those that can help to rectify the unexpected situation. Thus, consciousness solves the relevancy problem in recruiting resources.

7. **Setting goal context hierarchy.** Some responding behavior codelets instantiate an appropriate behavior stream, if a suitable one is not already in place. Using information from the conscious broadcast, they also bind variables and send activation to behaviors. (In our running example, a behavior stream to find jobs to offer Kevin Adams might be instantiated. His preference for Norfolk might be bound to a variable.) Here, we assume that there is such a behavior codelet and behavior stream. If not, then nonroutine problem solving using additional mechanisms is called for.[5]

8. **Action chosen.** The behavior net chooses a single behavior (goal context) and executes it. This choice may come from the just instantiated behavior stream or from a previously active stream. The choice is affected by

internal motivation (activation from goals), by the current situation, by external and internal conditions, by the relationship between the behaviors, and by the activation values of various behaviors. (In our example, IDA would likely choose to begin extracting useful data from Kevin Adam's personnel record in the Navy's database.)

9. **Action taken.** The execution of a behavior (goal context) results in the behavior codelets performing their specialized tasks, which may have external or internal consequences. This is IDA taking an action. The acting codelets also include an expectation codelet (see Step 6) with the task of monitoring the action and trying to bring to consciousness any failure in the expected results.

The IDA Conceptual Model

In addition to the IDA computational model that is now a successfully running software agent, several different additional capabilities for IDA have been designed and, sometimes, described. One felicitous feature of the IDA architecture is that it has been possible to design and add one new capability after another with literally no change in the basic processing structure of the system as outlined in the description of IDA's cognitive cycle just above. This makes us think we must be doing something right.

Some of the new capabilities are in the process of being implemented, while others await sufficient time, energy, and funding. This section is devoted to these new capabilities, including feelings and emotions, nonroutine problem solving, automization, transient episodic memory, metacognition, and several varieties of learning.

Feelings and Emotions

We view feelings and emotions as often suitable mechanisms for primary motivations (drives) in autonomous agents, including humans and many other animals (Sloman, 1987). Following Johnston, we conceive of emotions as special kinds of feelings—those with a necessary cognitive component (1999). While the computational IDA has directly implemented drives, IDA's predecessor, CMattie, had an implemented emotional apparatus (McCauley & Franklin, 1998). The newly designed feelings and emotions for IDA play a pervasive role in her entire cognitive process, as they do in humans (Damasio, 1999; Panksepp, 1998; Rolls, 1999). Here we will trace the many roles of feelings and emotions in the various steps of the cognitive cycle (Franklin & McCaulley, 2004).

In Step 1, IDA's perceptual memory, her slipnet, will have nodes for the various feelings, including emotions, as well as links connecting them to and from other nodes. Thus, the percept written to working memory in Step 2 may contain affective content. This affective content will serve as part of the cue used in Step 3 to retrieve local associations from transient episodic (see below) and declarative memories. The local associations so retrieved may contain accounts of affect associated with past events, as well as details of the event and of the action taken.

During the competition for "consciousness" in Step 4, attention codelets gathering information from long-term working memory will be influenced by affect. The stronger the affect, the higher the average activation of the coalition, and the more likely it is to win the competition. During the broadcast in Step 5, the various memories are updated, including the storing of affective information. In Step 6, various behavior codelets respond to the broadcast, and in Step 7, they instantiate behavior streams, bind variables, and activate behaviors. In addition to environmental activation, and based on feelings and emotions, these codelets will also activate behaviors that satisfy drives, thus implementing the drives. Those actions selected in Step 8 and performed in Step 9 are heavily influenced by the cumulative affect in the system.

Nonroutine Problem Solving

As humans, we have the ability to devise unexpected, and often clever, solutions to problems we have never before encountered. Sometimes they are even creative. According to global workspace theory, one principal function of consciousness is to recruit the resources needed for dealing with novel situations and for solving nonroutine problems. Though IDA's "consciousness" module is designed to deal intelligently with novel, unexpected, and problematic situations, the computational IDA is currently expected to handle only novel instances of routine situations. One message from a sailor asking that a job be found is much like another in content, even when received in natural language with no agreed upon protocol. Similarly, finding a new billet for one sailor will generally require much the same process as for another. Even the negotiation process between IDA and a sailor promises to be most frequently a relatively routine process. However, we expect IDA to occasionally receive messages outside of this expected group. Can IDA handle such a message intelligently by virtue of her "consciousness" mechanism alone? We do not think so. Additional mechanisms will be required.

One typical way for a nonroutine problem to arise is for some expectation to be unmet. I flip the switch, and the light does not turn on. In the IDA model, such an unexpected situation would likely be brought to "consciousness" by a particular type of attention codelet, an expectation codelet. In such a situation, behavior codelets responding in Step 6 may have no suitable behavior streams to instantiate (or one or more may be tried over several cycles and all fail). In this case, a behavior stream is instantiated that implements a variant of a partial-order planner (Gerevin & Schuber, 1996). This planner behavior stream operates in a deliberative (virtual) mode.

The start-state for the plan described here is the current state as reported by the contents of consciousness. The goal-state is chosen so as to satisfy the failed expectation. The backward-chaining planner uses as its initial set of operators the behavior codelets that responded to the broadcast during the cycle in which the planner behavior stream was implemented. The first (backward from the goal-state) step in the plan is chosen and is then written to working memory as IDA's action during the current cycle. On subsequent cycles, additional steps are added to the plan until the start-state is reached. The completed plan becomes a behavior stream that is saved and that is likely instantiated on some forthcoming cycle for trial. This process may be repeated as necessary. Nonroutine problem solving in IDA is a type of procedural learning. The collection of behavior stream templates, together with the behavior codelets, constitutes IDA's long-term procedural memory.

Automization

In this subsection, we briefly describe a mechanism by means of which a software agent can learn to automatize a skill that is a sequence of actions so as to perform it without "conscious" intervention (Negatu, McCauley, & Franklin, in review). Typically, when an action plan is learned for the first time, consciousness plays a major role. As the action plan is executed repeatedly, experience accumulates, and parts of the action plan eventually are automatized.

Motivated by pandemonium theory (Jackson, 1987), whenever any two codelets are active together in the IDA computational model, the association between them becomes stronger (or weaker if things are not going well). This is Hebbian learning (Hebb, 1949). Suppose an attention codelet, say AC1, belongs to the winning coalition, bringing with it the information that will eventually trigger the start of a given task. When the content of the coalition is broadcast to all behavior codelets, suppose a behavior codelet, say BC1, that responded instantiates a behavior stream appropriate for the task. IDA's behavior net eventually picks a particular behavior, say B1, for execution, and its underlying behavior codelets become active. For simplicity's sake, let us assume that each behavior in the stream has only one behavior codelet, and that B1 has its codelet, BC1, active.

If the action of BC1 attracts attention codelet AC2 and its coalition to "consciousness," then its content is broadcast. Similarly, suppose some behavior codelet, say BC2, under behavior B2 responds to this broadcast, and that B2 is chosen for execution, and BC2 becomes active.

Note that the codelets in the sequence BC1–AC2–BC2 are all overlappingly active together during a relatively short time period. Suppose our hypothetical task is executed repeatedly, producing the sequence BC1–AC2–BC2 repetitively. As automatization builds, the associations BC1–AC2, BC2–AC2, and BC1–BC2 increase. When the association BC1–BC2 is over threshold, the automatization mechanism allows BC1 to spread activation directly to BC2, causing it to become active without the intermediary of AC2. At the same time, the strong associations BC1–AC2 and BC2–AC2 diminish the attention codelet AC2's activation so that it has less probability to make it to "consciousness." Thus, the sequence BC1–AC2–BC2, which involves "consciousness," is transformed by IDA's automization mechanism into the subconscious action sequence BC1–BC2.

Transient Episodic Memory

Transient episodic memory is an unusual aspect of the IDA conceptual model. It is an episodic memory with a decay rate measured in hours. Though a "transient memory store" is often assumed (Panksepp, 1998, p. 129), the existence of such a memory has rarely been explicitly asserted (Donald, 2001; Conway, 2001; Baars & Franklin, 2003, Franklin et al., in review). In the IDA conceptual model, transient episodic memory is updated during Step 5 of each cognitive cycle with the contents of "consciousness." At this writing, we are in the process of expanding and testing our implementation of an experimental transient episodic memory using a ternary revision of sparse distributed memory allowing for an "I don't care" symbol (Ramamurthy, D'Mello, & Franklin, to appear).

Perceptual Memory

As described above, IDA's perceptual memory, including the slipnet and the perceptual codelets, is a fully implemented part of the running IDA computational model. The IDA conceptual model adds learning to this perceptual memory with updating during the broadcast (Step 5) of each cognitive cycle (Franklin et al., in review). New nodes and links are added to the slipnet as needed, while existing node and links have their base-level activations and weights updated, respectively.

Metacognition

IDA's predecessor, CMattie, had an impoverished metacognition module that prevented oscillations in her processing and tuned the parameters of her behavior net to make her more or less goal oriented or more or less opportunistic, etc. Metacognition in CMattie was implemented as a separate B-brain with its own decidedly different mechanism (Zhang, Dasgupta, & Franklin, 1998) that looked down on what the rest of CMattie was doing (Minsky, 1985) and interfered as needed.

Being dissatisfied with metacognition in CMattie, none was implemented in IDA. However, we have a back-burner project in mind to add a more powerful metacognitive capability to IDA using only her current architecture. This would be part of adding a reporting self to IDA, the subject of a different article. Metacognition would be implemented by a set of appropriate behavior codelets, behaviors, and behavior streams, together with suitable attention codelets.

Learning

The IDA model is also intended to learn in several different ways. In addition to learning via transient episodic and declarative memory as described above, IDA also learns via Hebbian temporal association, as discussed in the section on automization. A coalition that comes to "consciousness" substantially increases the associations between the codelets that form the coalition. The same is true, to a lesser extent, when they are simply active together. Recall that these associations provide the basis coalition formation.

Step 5 of the cognitive cycle, the broadcast step, also describes the updating of IDA's perceptual memory, the slipnet, using the contents of "consciousness." Procedural learning in the IDA conceptual model also occurs during Step 5, with "conscious" contents providing reinforcement to actions of behaviors. These two forms of learning use a similar mechanism, a base-level activation, the first for nodes (and weights on links), and the second for primitive behavior codelets.

Yet another form of learning in the IDA conceptual model is chunking. The chunking manager gathers highly associated coalitions of codelets into a single supercodelet in the manner of concept demons from pandemonium theory (Jackson, 1987) or of chunking in SOAR (Laird et al., 1987).

Hypotheses

Each design decision as to architecture and mechanisms taken in constructing the IDA conceptual model translates directly into a hypothesis about human cognition for cognitive scientists and neuroscientists (Franklin, 1997). In this section, we highlight a few of these hypotheses (Franklin et al., in review):

1. **The cognitive cycle.** Much of human cognition functions by means of cognitive cycles, continual interactions between conscious content, various memory systems, and the action selection mechanism. The IDA model suggests that consciousness occurs as a sequence of discrete, coherent episodes separated by short periods of no conscious content (see also, VanRullen & Koch, 2003).
2. **Transient episodic memory.** Humans have a content-addressable, associative, transient episodic memory with a decay rate measured in hours (Conway, 2001). In our theory, a conscious event is stored in transient episodic memory by a conscious broadcast. A corollary to this hypothesis says that conscious contents can only be encoded (consolidated) in long-term declarative memory via transient episodic memory.
3. **Perceptual memory.** A perceptual memory, distinct from semantic memory but storing much the same contents, exists in humans and plays a central role in the assigning of interpretations to incoming stimuli. The conscious broadcast begins and updates the processes of learning to recognize, to categorize, and to form concepts, all employing perceptual memory.
4. **Procedural memory.** Procedural skills are shaped by reinforcement learning operating through consciousness over more than one cognitive cycle.
5. **Voluntary and automatic attention.** In the Global Workspace/IDA model, attention is the process of bringing contents to consciousness. Automatic attention occurs subconsciously and without effort during a single cognitive cycle. Attention may also occur voluntarily in a consciously goal-directed way, over multiple cycles.

Conclusion

Here I hope to have described an architecture capable of implementing many human cognitive, including affective, functions within the domain of an autonomous software agent. I would hesitate to claim that this architecture, as is, is fully functioning by human standards. Even the conceptual model lacks, for instance, the typical human senses of vision, olfaction, audition, etc. Its contact with the world is only through strings of characters. There is only the most rudimentary sensory fusion by the agents. They lack selves, self-awareness, and the ability to report internal events. There is much work left to be done.

Nonetheless, the IDA conceptual model, together with its architecture and mechanisms, does answer, for the agent, the question of what to do next. Thus, it constitutes a theory of mind. The IDA model seems to satisfy the requirements of an artificial mind as outlined in the introduction above. Something of this view may also be ascribed to Owen Holland, who wrote as follows (2003):

> *In many ways, Stan Franklin's work on "conscious software" offers a real challenge to functionalists. If consciousness is what consciousness does, then his systems may well exceed the requirements, in that they not only mimic successfully the outcomes of conscious processes in some humans (naval dispatchers [sic]) but they do it in the way that the conscious human brain appears to do it, since their functional components are explicitly modeled on the elements of Baars' global workspace theory.... (p. 3)*

Acknowledgments

This research was supported in part by ONR grant N00014-98-1-0332 and was produced with essential contributions from the "Conscious" Software Research Group, including Bernard Baars, Art Graesser, Lee McCauley, Satish Ambati, Ashraf Anwar, Myles Bogner, Sidney D'Mello, Arpad Kelemen, Ravikumar Kondadadi, Irina Makkaveeva, Aregahegn Negatu, Hongjun Song, Allexei Stoliartchouk, Uma Ramamurthy, Matthew Ventura, and Zhaohua Zhang.

References

Allen, J. J. (1995). *Natural language understanding*. Redwood City, CA: Benjamin/Cummings.

Anwar, A., Dasgupta, D., & Franklin, S. (1999). Using genetic algorithms for sparse distributed memory initialization. *International Conference Genetic and Evolutionary Computation (GECCO)*. July 6–9.

Anwar, A., & Franklin, S. (2003). Sparse distributed memory for "conscious" software agents. *Cognitive Systems Research, 4*, 339–354.

Baars, B. J. (1988). *A cognitive theory of consciousness*. London; Oxford: Cambridge University Press.

Baars, B. J. (1997). *In the theater of consciousness*. Oxford: Oxford University Press.

Baars, B. J. (2002). The conscious access hypothesis: Origins and recent evidence. *Trends in Cognitive Science, 6*, 47–52.

Baars, B. J., & Franklin, S. (2003). How conscious experience and working memory interact. *Trends in Cognitive Science, 7*, 166–172.

Baddeley, A. D. (1993). Working memory and conscious awareness. In A. Collins, S. Gathercole, M. A. Conway, & P. Morris (Eds.), *Theories of memory*. Mahweh, NJ: Lawrence Erlbaum.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22,* 577–609.

Bogner, M. (1999). *Realizing "consciousness" in software agents*. PhD dissertation. University of Memphis, TN.

Bogner, M., Ramamurthy, U., & Franklin, S. (2000). "Consciousness" and conceptual learning in a socially situated agent. In K. Dautenhahn (Ed.), *Human cognition and social agent technology*. Amsterdam: John Benjamins.

Conway, M. A. (2001). Sensory-perceptual episodic memory and its context: Autobiographical memory. In A. Baddeley, M. Conway, & J. Aggleton (Eds.), *Episodic memory*. Oxford: Oxford University Press.

Damasio, A. R. (1999). *The feeling of what happens*. New York: Harcourt Brace.

Donald, M. (2001). *A mind so rare*. New York: Norton.

Edelman, G. M. (1987). *Neural Darwinism*. New York: Basic Books.

Ericsson, K. A., & Kintsch, W. (1995). Long-term working memory. *Psychological Review, 102*, 21–245.

Franklin, S. (1995). *Artificial minds*. Cambridge, MA: MIT Press.

Franklin, S. (1997). Autonomous agents as embodied AI. *Cybernetics and Systems, 28*, 499–520.

Franklin, S. (2001). Conscious software: A computational view of mind. In V. Loia & S. Sessa (Eds.), *Soft computing agents: New trends for designing autonomous systems*. Berlin: Springer (Physica-Verlag).

Franklin, S., Baars, B. J., Ramamurthy, U., & Ventura, M. (In review). The role of consciousness in memory.

Franklin, S., & Graesser, A. C. (1997). Is it an agent, or just a program?: A taxonomy for autonomous agents. In *Intelligent Agents III*. Berlin: Springer Verlag.

Franklin, S., & Graesser, A. (1999). A software agent model of consciousness. *Consciousness and Cognition, 8*, 285–305.

Franklin, S., & Graesser, A. (2001). Modeling cognition with software agents. In J. D. Moore & K. Stenning (Eds.), *CogSci2001*: *Proceedings of the 23rd Annual Conference of the Cognitive Science Society*, August 1–4. Mahwah, NJ: Lawrence Erlbaum Associates.

Franklin, S., Kelemen, A., & McCauley, L. (1998). IDA: A cognitive agent architecture. In *IEEE Conference on Systems, Man and Cybernetics*. Washington, DC: IEEE Press.

Franklin, S., & McCaulley, L. (2004). Feelings and emotions as motivators and learning facilitators. *Architectures for Modeling Emotions. AAAI Spring Symposia Technical Series* [Technical Report SS-04-02].

Gerevin, A., & Schuber, L. (1996). Accelerating partial-order planners: Some techniques for effective search control and pruning. *Journal of Artificial Intelligence Research, 5*, 95–137.

Hebb, D. O. (1949). *Organization of behavior*. New York: John Wiley.

Hofstadter, D. R., & Mitchell, M. (1994). The Copycat Project: A model of mental fluidity and analogy-making. In K. J. Holyoak & J. A. Barnden (Eds.), *Advances in connectionist and neural computation theory, Vol. 2: Logical connections*. Norwood NJ: Ablex.

Holland, J. H. (1986). A mathematical framework for studying learning in classifier systems. *Physica, 22 D*, 307–317. (Also in J. D. Farmer, A. Lapedes, N. H. Packard, & B. Wendroff [Eds.], *Evolution, games and learning*. Amsterdam: Elsevier/North Holland.)

Holland, O. (Ed.). (2003). Special issue on machine consciousness. *Journal of Consciousness Studies, 10*(4–5).

Jackson, J. V. (1987). Idea for a mind. *Siggart Newsletter, 181*, 23–26.

Johnson, V. S. (1999). *Why we feel: The science of human emotions*. Reading, MA: Perseus Books.

Kanerva, P. (1988). *Sparse distributed memory*. Cambridge, MA: The MIT Press.

Kelemen, A., Liang, Y., Kozma, R., & Franklin, S. (2002). Optimizing intelligent agent's constraint satisfaction with neural networks. In A. Abraham & B. Nath (Eds.), *Innovations in intelligent systems*. Heidelberg: Springer-Verlag.

Laird, E. J., Newell, A., & Rosenbloom, P. S. (1987). SOAR: An architecture for general intelligence. *Artificial Intelligence, 33*, 1–64.

Maes, P. (1989). How to do the right thing. *Connection Science, 1*, 291–323.

Maturana, H. R. (1975). The organization of the living: A theory of the living organization. *International Journal of Man–Machine Studies, 7*, 313–332.

Maturana, R. H., & Varela, F. J. (1980). *Autopoiesis and cognition: The realization of the living, Dordrecht*. Netherlands: Reidel.

McCauley, T. L., & Franklin, S. (1998). An architecture for emotion. In *AAAI Fall Symposium Emotional and Intelligent: The Tangled Knot of Cognition*. Menlo Park, CA: AAAI Press.

Minsky, M. (1985). *The society of mind*. New York: Simon & Schuster.

Negatu, A., & Franklin, S. (2002). An action selection mechanism for "conscious" software agents. *Cognitive Science Quarterly, 2*, 363–386.

Negatu, A., McCauley, T. L., & Franklin, S. (In review). Automatization for software agents.

Ornstein, R. (1986). *Multimind*. Boston, MA: Houghton Mifflin.

Panksepp, J. (1998). *Affective neuroscience*. Oxford: Oxford University Press.

Picard, R. (1997). *Affective computing*. Cambridge, MA: The MIT Press.

Prem, E. (Ed.). (1997). Epistemological aspects of embodied artificial intelligence. *Cybernetics and Systems, 28*(5&6).

Ramamurthy, U., D'Mello, S., & Franklin, S. (to appear). Modified sparse distributed memory as transient episodic memory for cognitive software agents. In *Proceedings of the International Conference on Systems, Man and Cybernetics*, Piscataway, NJ: IEEE.

Rolls, E. T. (1999). *The brain and emotion*. Oxford: Oxford University Press.

Shastri, L. (2001). A computational model of episodic memory formation in the hippocampal system. *Neurocomputing, 38–40*, 889–897.

Shastri, L. (2002). Episodic memory and cortico-hippocampal interactions. *Trends in Cognitive Sciences, 6*, 162–168.

Sloman, A. (1987). Motives mechanisms emotions. *Cognition and Emotion, 1*, 217–234.

Sloman, A. (1998). The "semantics" of evolution: Trajectories and trade-offs in design space and niche space. In H. Coelho (Ed.), *Progress in artificial intelligence*. Berlin: Springer.

Sloman, A., & Chrisley, R. (2003). Virtual machines and consciousness. *Journal of Consciousness Studies, 10*, 133–172.

Song, H., & Franklin, S. (2000). A behavior instantiation agent architecture. *Connection Science, 12*, 21–44.

Taylor, J. G. (1999). *The race for consciousness*. Cambridge, MA: MIT Press.

Valenzuela-Rendon, M. (1991). The fuzzy classifier system: A classifier system for continuously varying variables. In *Proceedings of the Fourth International Conference on Genetic Algorithms*. San Mateo, CA: Morgan Kaufmann.

VanRullen, R., & Koch, C. (2003). Is perception discrete or continuous? *Trends in Cognitive Science, 7*, 207–213.

Zhang, Z., Dasgupta, D., & Franklin, S. (1998). Metacognition in software agents using classifier systems. In *Proceedings of the 15th National Conference on Artificial Intelligence*. Madison, WI: MIT Press.

Zhang, Z., Franklin, S., Olde, B., Wan, Y., & Graesser, A. (1998). Natural language sensing for autonomous agents. In *Proceedings of IEEE International Joint Symposia on Intellgence Systems ’98*.

Endnotes

[1] My use of feminine pronouns when speaking of IDA simply seems more comfortable given that she seems independent, intelligent, and capable, and that the acronym is a common English feminine name. Nothing more should be read into it.

[2] Not to be confused with the phrase “theory of mind” as used by ethologists.

[3] The term “cognitive” is used here in a broad sense so as to include affect.

[4] We make no claim that IDA is phenomenally conscious. On the other hand, we know of no convincing argument that she is not.

[5] Though part of the IDA conceptual model, the material on nonroutine problem solving has not yet been published.

Chapter 9

The Integration and Control of Behaviour: Insights from Neuroscience and AI

David W. Glasspool
Cancer Research UK, London, UK

Abstract

Clues to the way behaviour is integrated and controlled in the human mind have emerged from cognitive psychology and neuroscience. The picture that is emerging mirrors solutions (driven primarily by engineering concerns) to similar problems in the rather different domains of mobile robotics and intelligent agents in artificial intelligence (AI). This chapter looks in detail at the relationship between a psychological theory of willed and automatic control of behaviour, the Norman and Shallice framework, and three types of engineering-based theory in AI. As well as being a promising basis for a large-scale model of cognition, the Norman and Shallice framework presents an interesting example both of apparent theoretical convergence between AI and empirical psychology, and of the way in which theoretical work in both fields can benefit from interaction between them.

Introduction

Building a computationally specified theory of a human-level mind is an ambitious goal. How can the cognitive disciplines—artificial intelligence (AI) and cognitive psychology—contribute to such an undertaking? Both psychology and AI have tended to study small areas of cognition and work with theories of single empirical phenomena. In a full-scale cognitive theory, two related issues must be addressed, those of integration (how are numerous cognitive theories or models organised into a coherent whole, rather than descending into behavioural chaos?) and control (how are these modules to be coordinated by an overall goal?).

Within cognitive psychology, few theories have emerged that attempt to deal with this breadth of human cognitive function. One that does is the framework proposed by Norman and Shallice (1986). In the context of the research goal mentioned above, this theory is interesting for a number of reasons.

First, it takes the form of a layered architecture. There is a strong parallel with certain forms of layered agent architecture that have been developed within AI, although the rationale for the architectures and the scientific tradition from which they have emerged are different.

Second, the elements of the Norman and Shallice framework relate closely to analogous theories or models in AI. For example, the middle layer of the framework is similar to AI models of action selection. Thus, there is potential for useful cross-fertilisation of ideas between comparable theories in the two disciplines.

Finally, the various elements of the Norman and Shallice framework are now sufficiently well specified that computational implementations have been produced for the major components of the framework.

The next section discusses the first of these points—the correspondence between the three levels of control in the Norman and Shallice framework and a class of AI agent models. It begins by introducing the framework and the empirical phenomena on which it is predicated. The framework and its rationale are then compared with a group of three-layer architectures in AI that are motivated largely on engineering grounds. We then move on to examining the internal operation of the main elements of the Norman and Shallice model. The middle layer of the framework is compared with an action selection model in AI. The upper level is compared with an agent architecture model from AI—the *Domino* model. Finally, the "Discussion" section notes some connections with other approaches in the literature, reviews the work that has been done toward a computational implementation of the Norman and Shallice framework, and discusses the future prospects for the general architecture identified in the chapter.

Three-Layer Agent Architectures in Psychology and AI

Within cognitive psychology and AI, there have been two approaches to large-scale theories of cognition, which can be characterised as *unified* and *modular*. A unified model is one in which a single mechanism is responsible for the majority of processes involved in cognition, for example, the unified cognitive models of Newell (1990) and Anderson and Lebiere (1998), and *pipelined* architectures (Nilsson, 1984). A modular model is one in which cognition is emergent over multiple processes operating in parallel, each contributing to the overall capabilities of the agent (for example, Minsky, 1986; Brooks, 1991). Over recent years, modular theories within AI have emerged as strong challengers to earlier unified theories. Cognitive psychology has not produced so many large-scale theories, but a similar debate has gone on between the two approaches (for example, Cooper & Shallice, 1995).

This section introduces and compares two "cognitive architecture" theories originating from within cognitive psychology and AI which take the second, modular approach. The psychological theory, the Norman and Shallice framework (Norman & Shallice, 1986), is selected because it is a well-known example of the few large-scale modular psychological theories. The AI theory, a class of three-layered agent architectures, is taken as representative of a successful area of agent research in AI and is selected because it shows interesting similarities with the Norman and Shallice framework in its approach.

Norman and Shallice Framework for Behaviour Control

The Norman and Shallice framework for willed and automatic action encompasses the automatic control of habitual action and its adaptation to changing circumstances and the deliberate control of and monitoring of "higher level" behaviour. The useful evidence for a psychological theory of this sort comes not so much from observing what people are capable of but from the ways in which the mechanism can break down. The theory is informed by the slips and errors of everyday action in normal individuals and by the characteristic errors made by patients with a variety of neurological disorders. These types of errors can provide rich diagnostic information concerning the types of mechanisms responsible. We will briefly review these before outlining the theory.

Action Lapses and Slips

Much of our everyday behaviour is so well learned that we can carry it out automatically. Getting dressed, driving a car, and conducting similar activities can often be carried out with little conscious attention. However, we all occasionally make errors in such behaviour. Reason (1984) studied the slips and lapses made by normal individuals during routine behaviour. Errors turn out to be surprisingly common but can be classified as belonging to a limited set of types. These include errors of place substitution (for example, putting the kettle, rather than the milk, into the fridge after making coffee), errors of object substitution (for example, opening a jar of jam, not the coffee jar, when intending to make coffee), errors of omission (for example, pouring water into a teapot without boiling it), and errors involving the capture of behaviour by a different routine (such as going upstairs to get changed but getting into bed). Reason finds that the situations in which such slips and lapses occur share two properties: the action being performed is well learned and routine, and attention is distracted, either by preoccupation or by some external event.

There are two points of interest here. First, it is clear that we can perform a wide range of often complex habitual actions without concentrating on them—the control of well-learned action can become automatic and does not apparently need conscious supervision. Second, when we allow such behaviour to proceed without our conscious control, it is susceptible to a specific range of characteristic errors. These observations provide one class of data that psychological theories of action control must address. Another important class of data is provided by the effects of neurological damage.

Neurological Impairment of Behaviour Control

Cooper and Shallice (2000) reviewed a range of problems with the control of action following neurological damage (mainly to areas of prefrontal cortex). Here, we briefly mention three syndromes of particular interest.

Patients with action disorganisation syndrome (ADS; Schwartz et al., 1991; Humphreys & Forde, 1998) make errors similar in type to those of normal individuals—errors in the sequencing of actions, the omission or insertion of actions, or the substitution of place or object. However, their errors are far more frequent. For example, patient HH of Schwartz et al. (1991) made 97 such errors during 28 attempts at making a cup of coffee.

Utilisation behaviour (Lhermitte, 1983) can be characterised as weakening of intentional control of action, so that irrelevant responses suggested by the environment may take control of behaviour. A patient may pick up and perform actions with items lying on a table, for example, which are appropriate to the items but not relevant to the task in hand.

Patients with "strategy application disorder" (Shallice & Burgess, 1991) are able to carry out individual tasks but have difficulty coordinating a number of simultaneous task demands. For example, they may be able to carry out individual food preparation tasks but are unable to plan and cook a meal. Their deficit appears to be in the ability to schedule multiple tasks over an extended period.

These data again point to the idea that certain well-learned or habitual tasks can be carried out more or less autonomously, but supervision (by some system that can be independently damaged) is required to produce behaviour with more complex or high-level organisation.

The Norman and Shallice Framework

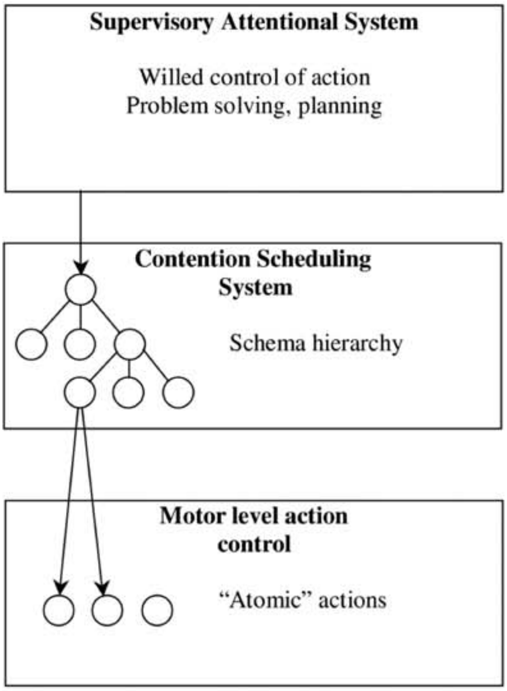
The challenge for a psychological account of the integration and control of behaviour is to explain data of the types outlined above. In the earliest version of the theory, Norman and Shallice (1986) interpreted the data as implying that two distinct systems operate to control everyday behaviour. An *automatic* system coordinates well-learned habitual behaviour that is assumed to make up much of our behavioural repertoire. This system, *Contention Scheduling* (CS), comprises a set of action schemas, each of which encodes a behavioural sequence. Schemas compete for control of behaviour under the influence of stimulus from the environment and of top-down control from the second system, a *Supervisory Attentional System* (SAS). The SAS is responsible for willed or attentional control of action. It sets high-level goals for behaviour that CS carries out autonomously, it can generate new action schemas for CS in novel situations, it monitors behaviour, and it can intervene to take direct control of behaviour in critical situations. The SAS can influence the CS system but has no direct access to motor control. Roughly, these two subsystems separate along the lines introduced at the start of the chapter: CS is responsible for integrating disparate facets of behaviour into a coherent whole, while SAS is responsible for setting and pursuing high-level goals.

More recently, Cooper and Shallice (2000) provided a number of arguments for adding a further level of behavioural control below the CS system. This lower level, *motor behaviour*, comprises the individual motor commands required to carry out a simple action (extending and retracting individual muscle groups, for example). The CS component operates with actions at the lowest level to which they are referred in everyday language—grasping, reaching, etc.—which abstract over more complex motor-level actions. The motor control system is capable only of relatively simple actions but closely couples sensory input with motor control to produce actions that are well tailored to their context. For example, given the command to grasp an object, it coordinates the operation of various muscle groups to tailor the action to the location and configuration of the object. The three layers of this extended framework are shown in Figure 1.

Looking in more detail at the framework, the contention scheduling system comprises a hierarchy of action schemas. Schemas may represent discrete actions or may organise sequences of actions (or sequences of other schemas). The schema hierarchy terminates in a set of motor-level actions that are held to be carried out directly by motor systems. Actions at this level might include, for example, "pick up an item," "unscrew," or "stir." Higher-level schemas might include "open jar," which would organise the actions of picking up, unscrewing a lid, and putting it down. At a higher level still, a "make coffee" schema might exist.

Schemas are connected in an interactive-activation network. They are activated from the top-down by their parent schemas or by control from the SAS, and from the bottom-up by input from the environment. They compete for execution on the

Figure 1. Norman and Shallice's (1986) framework for action control augmented with Cooper and Shallice's (2000) distinction between cognitive and motor level action


Supervisory Attentional System
Willed control of action
Problem solving, planning
Contention Scheduling System
Schema hierarchy
Motor level action control
"Atomic" actions

basis of their activation level. A schema is triggered when its activation level is higher than any other schema and higher than a trigger threshold. A triggered schema feeds activation forward to its child schemas. Top-down activation can exert detailed control over behaviour, or it can simply be used to specify goals, by activating high-level schemas. Such schemas may provide multiple ways for a goal to be achieved—coffee can be supplied in a jar or a packet, for example, so a schema for adding coffee to a mug can be indifferent to the particular lower-level behaviour required to achieve its goal. Whichever suitable subschema best fits the current configuration of the environment will automatically be selected.

Cooper and Shallice (1997, 2000) simulated the CS system in detail. With a certain amount of background noise in the system, and a reduction in top-down input, the system makes occasional errors analogous to those made by normal individuals, when the wrong schema or subschema is triggered. By varying the parameters of the model, utilisation behaviour and ADS can be simulated, as well as a number of other neuropsychological disorders of action control.

The SAS exerts control by directly activating individual low-level actions or by causing the triggering of an existing schema that would not otherwise be triggered in that situation. Internally, however, the SAS is poorly specified. Based largely on neuropsychological evidence but partially guided by a priori reasoning about the types of processes that must be involved in supervisory processing, Shallice and Burgess (1996) outlined the processes involved in the SAS and their relationships during supervisory processing. They characterised the functioning of the SAS as centrally involving the construction and implementation of a temporary new schema, which can control lower-level CS schemas so as to provide a procedure for dealing effectively with a novel situation.

The Organisation of Action in Mobile Robotics

Problems of integration and control of behaviour have also been explored in a number of areas of AI research. One area in which these issues have been particularly important is mobile robotics. Early robotics projects (for example, Nilsson, 1984; Moravec, 1982) employed architectures centering on classical planning systems, but robots based on this paradigm tend to be slow, cumbersome, and fragile in their operation. In the mid-1980s, an alternative approach began to emerge, typified by the work of Brooks (1986) and Kaelbling (1990), that does away with a central representation of the world and uses many simple, high-speed processes coupling simple sensory systems directly to action, each implementing small, circumscribed elements of behaviour. The approach allows small, fast, robust, and flexible robot control systems to be built, and it is often referred to as "reactive planning," or more generally, as behaviour-based AI (BBAI).

Brooks' work, in particular, gave rise to rapid theoretical development. It became apparent that a *pure* reactive approach suffers from some shortcomings in practical applications. Primarily, there is no encapsulation of existing behaviours to allow them to be used as primitives by new ones. One consequence is that it becomes increasingly difficult to program the system as more complex behaviour patterns are attempted (Hartley & Pipitone, 1991). Maes (1991) also pointed out the difficulty of implementing higher-order cognitive functions, such as planning.

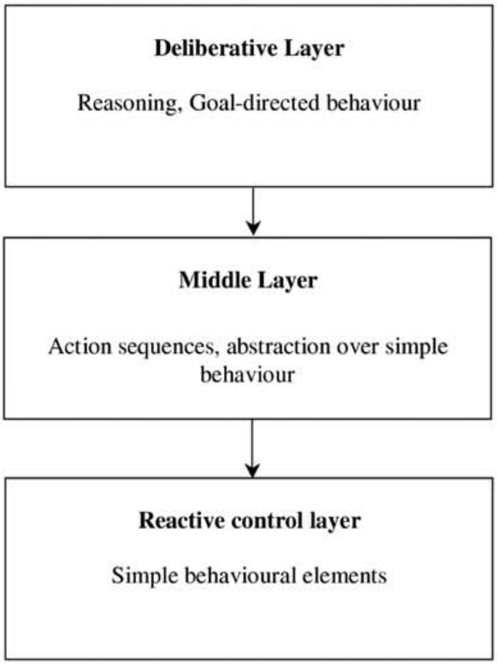
A second consequence of the lack of abstraction of lower-level behaviours is the difficulty of *tasking* a reactive controller (Gat, 1998). The lack of clean abstraction boundaries between behavioural subsystems means it is difficult to program the architecture to carry out a new task, short of completely redesigning it. This also makes it difficult for such architectures to be built to learn new behavioural routines from experience (Maes, 1991).

A third problem results from the emphasis on avoiding the storage of any persistent information about the environment. This is motivated by the need for the control system to remain highly reactive, but reducing persistent state throughout the architecture means that it is difficult to follow extended plans (Gat, 1998) or to arrange for multiple long-range goals to guide behaviour (Hexmoor & Kortenkamp, 1995).

Much subsequent work has concentrated on providing a means for encapsulating useful chunks of reactive behavioural control. A number of different initiatives converged on a three-layered organisation (Connell, 1991; Gat, 1991; Bonasso, 1991; Elsaessar & Slack, 1994; Firby, 1994; Bonasso, Kortenkamp, Miller, & Slack, 1997). Hexmoor and Kortenkamp (1995) pointed out an emerging consensus among a group of agent architectures on the identity of the three layers (Figure 2). A bottom layer comprising a Brooks-style reactive controller provides flexible and robust low-level control. A deliberative high-level system provides more traditional AI planning and problem-solving capability. The middle layer interfaces between the two. It provides abstractions over lower-level behaviours in two ways: by constructing more powerful behavioural elements through assembling sequences of simple behaviours, and by providing higher-level goals that may be achieved by different lower-level actions, depending on prevailing circumstances.

Gat (1998) defined the boundaries between these layers in terms of stored state. The lower-level controller stores no long-term state. It is able to respond quickly and cleanly to unexpected changes in the state of the environment. The middle layer stores state concerning the past. This allows it to work through planned sequences of actions, to identify failures to achieve intermediate goals, and to take simple remedial action. The top level represents information about the past and the future. It can reason about long-term goals and generate plans that can achieve them. While defining the layers in these terms, however, Gat pointed out

Figure 2. Three-layer architecture of Gat (1998) and others


Deliberative Layer
Reasoning, Goal-directed behaviour
Middle Layer
Action sequences, abstraction over simple behaviour
Reactive control layer
Simple behavioural elements

that the decomposition is motivated empirically, from the observation that algorithms for robot control tend to fall into three distinct classes: stateless control procedures of the Brooks type; algorithms for combining lower-level behaviours into flexible precompiled sequences; and algorithms for time-consuming search-based procedures, like planning and reasoning.

For convenience, the style of three-layer architecture outlined here will be referred to as "TLA" (for three-layer architecture). It should be noted that other types of layered organisation have also been proposed, for example, Sloman (2002). The Gat-style TLA framework has, however, been particularly successful in controlling implemented AI agents. For example, NASA used the approach in an autonomous space probe (Muscettola et al., 1998).

Convergent Architectures?

The Norman and Shallice framework and the TLA framework address similar issues of control and integration of an agent's behaviour in two different domains. Comparing Figures 1 and 2, the correspondence between the two is striking. Might the resemblance simply be superficial? We need to compare the ways the layers are specified in each approach.

Shallice and Burgess (1996) described the SAS as corresponding to frontal-lobe processes "critically involved in coping with novel situations as opposed to routine ones" (p. 1406). They specified its functions in terms of goal setting, problem solving, and schema generation (planning). Gat (1998) described the topmost TLA system as "the locus of time-consuming computations. Usually this means such things as planning and other exponential search-based algorithms... It can produce plans for the [middle layer] to implement, or it can respond to specific queries from the [middle layer]." In other words, the main functions are generating new plans of action and dealing with novel situations. Despite the language differences—an inevitable consequence of comparison across disciplines—the two architectures apparently ascribe essentially the same functions to their highest level systems.

Turning to the lowest level of behaviour control, on Cooper and Shallice's (2000) account, this corresponds to motor-level actions. These operations are the preserve of motor systems and are not susceptible to the types of errors typically made at the cognitive level. Cooper and Shallice (2000) thus made an empirical distinction between the lowest (motor) level and middle (CS) level.

Gat (1998) described the processes at the lowest TLA level as "designed to produce simple primitive behaviours that can be composed to produce more complex task-achieving behaviour." The composition of simple behaviours into complex behaviour is a function of the middle layer. There is no theoretically principled line drawn between simple behaviours and complex ones; rather, Gat provided a number of guidelines for making the distinction. For example, behavioural control at the reactive level should keep internal state to a minimum and use only input-output transfer functions that are continuous with respect to internal state.

The notion of reactive control (tight sensory-to-motor coupling) is an important part of the TLA definition of this layer. The triggering of action by environmental input is not stressed in Cooper and Shallice's characterisation of the motor level, and the notion of reactive behaviour control, in the BBAI sense, is not prominent in the psychological literature. However, reflex and sensory-motor feedbacks are known to play an important part in low-level human motor control. [Interestingly, this type of control is part of the definition of the CS system. Cooper and Glasspool (2001), for example, treat the environmental triggering conditions of schemas in CS as affordances for action, priming appropriate behaviour in response to learned environmental configurations.]

The common ground between the two approaches' view of the lowest level of action control is that it is the domain of *atomic* actions, not further decomposable without getting into detail that is irrelevant to a task-centred view of action, and that atomic actions are strongly shaped by direct environmental input. To the extent that they place a task-centered view of action control above this level, and the non-task-specific detail required to achieve it within the lowest layer, the characterisations of this layer can be viewed as being at the same conceptual level.

In the TLA account, a primary function of the middle layer is to organise primitive behaviours into behaviour sequences that perform two functions: they form a more compact and convenient representation of behaviour for use by higher level processes (sequences of behaviour that are often needed are chunked together), and they allow alternative means to be specified for achieving a goal, providing low-level flexibility and avoiding the need to specify behaviour in detail. Both of these functions are central to the Norman and Shallice CS system. Schemas represent well-learned fragments of behaviour and provide a goal-based representation—subschemas for achieving the same goal compete to service a higher-order schema's requirements. Functionally, the CS corresponds well to the TLA middle layer.

There seems to be more than a superficial resemblance between the two architectures. Why might this particular arrangement of layers have emerged in both approaches? The separation of high-speed reactive processes that allow an agent to respond in a timely way to a dynamic environment from slow, resource-intensive deliberative processes that require stored representations makes good engineering sense. Assigning these functions to separate layers of the control system implies that each covers the full breadth of the action control problem, from sensory input to motor output, and that both subsystems operate in parallel. The relatively simple processes required to keep an agent safe and correctly oriented in its environment can operate at full speed, regardless of the amount of deliberation going on concurrently. The approach typified by Gat's architecture is not the only one to recognise the benefits of this arrangement. For example, Sloman (2002) had separate reactive and deliberative layers operating in parallel.

The distinctive feature of the Gat approach, which is shared by the Norman and Shallice framework, is the role of the middle layer of control in mediating between these two extremes. Hexmoor and Kortenkamp (1995) referred to the middle layer in this arrangement as providing a *differential* between the different processing styles of the two outer layers. In both the Gat and the Norman and Shallice approaches, this differential takes the form of an abstraction layer. The middle layer is concerned with two problems: generating flexible sequences of action that further the agent's goals, while accommodating the particular environment in which the actions are being carried out. It achieves these goals by providing two services: It allows complex behavioural routines to be built up from simpler units, allowing the higher level deliberative controller to ignore the details of complex actions that have been prelearned; and it allows alternative low-level behaviours to be provided for achieving a single goal, allowing the deliberative layer to ignore the details of adapting behaviour to circumstances. These services can be viewed as two distinct ways in which abstraction of lower levels is provided for higher levels of the schema hierarchy: *Compositional abstraction* allows complex actions to be built up from sequences of simpler actions. *Implementational abstraction* provides alternative means of achieving a higher level goal.

In terms of addressing the limitations of *pure* BBAI identified above, compositional abstraction allows the reactive control system to be *tasked* by setting goals at runtime (rather than having goals implicit in the design of the reactive controller). Implementational abstraction allows the operation of the system in achieving these goals to flexibly adapt to a dynamic environment.

The engineering rationale for splitting these abstraction functions into a layer of their own stems from their distinct computational status. They are not reactive behaviours; they are abstractions over reactive behaviours, and particularly in the case of compositional abstractions, they need to be persistent rather than reactive in the face of a changing environment (so that sequences of behaviour can be pursued, even though the environment is changing). They are also not generally deliberative. They provide convenient chunks of behaviour as the raw material for deliberative-level plans, rather than carrying out planning themselves. Compositional abstractions (behavioural sequences) need to persist regardless of the computational load on the deliberative system. A separate control system running in parallel with the upper and lower layers and providing this intermediate level of control appears well motivated on engineering grounds.

We look in more detail at the abstraction functions of the middle layer of the Norman and Shallice framework in the next section, where we compare it with a different attempt to add structure and abstraction to Brooks' BBAI approach.

Norman and Shallice Framework Compared with Maes' Approach to Action Organisation

A well-known attempt to extend the pure BBAI approach to provide higher-order cognitive capabilities is Maes' (1991) Agent Network Architecture (ANA). Maes built a spreading activation network of Brooks-style behaviour modules and added an action selection mechanism. The approach has many points of similarity with the CS component of the Norman and Shallice framework and to related models of action selection in the AI literature (Tyrell, 1993; see also Bryson, 2000). Also of interest is the fact that Maes' ANA has been criticised as unsuitable for controlling practical agents. We look in this section at the relationship between the two approaches and at the criticisms of ANA and their applicability to CS.

The previous section identified compositional and implementational abstraction as the twin functions of the middle layer of the Gat and Norman and Shallice architectures. In Cooper and Shallice's (2000) implementation of CS, these two types of abstraction are achieved by a goal/action hierarchy. The CS system comprises alternating layers of schema nodes and goal nodes. Each goal node represents a goal that can be achieved automatically by the CS system, and it is parent to one or more schema nodes representing alternative methods for achieving that goal, providing implementational abstraction over the different methods. An automatic action-selection mechanism determines which schema node will be used to attempt to attain the goal. Alternative schemas that can achieve the same goal compete on the basis of their activation level, which is influenced by top-down input from the SAS and by the state of the environment, so methods of achieving goals are tailored to the objects available in the environment and the actions they afford.

Similarly, each schema node in the CS system is parent to a set of goal nodes representing the subgoals that must be achieved in order to carry out the schema, providing compositional abstraction over sequences of actions. The ordering of subgoals beneath a schema is achieved via preconditions on each action, which impose conditions on what must be true in the world for the action to take place.

Maes' ANA shares several aims with CS. It too is concerned with generating sequences of actions that further high-level goals and accommodate the changing state of the world. The primary aim of ANA is to accommodate persistent goals and dynamic planning within a pure BBAI framework. It treats Brooks-style reactive behaviours as atomic modules that are each given a real-valued activation level and are connected via bidirectional links along which activation can flow.

Like schemas in CS, modules in ANA have preconditions that are states of the environment. However, they are connected by links to other modules that tend to establish those states, so modules tend to be organised into sequential chains rather than hierarchically. Activation can flow forward along these chains when a module is activated, which can happen when the environment is in a state that satisfies its preconditions. Activation then flows to any modules that have preconditions likely to be satisfied when the module executes. Activation also flows backwards from modules that represent goals of the agent. These tend to activate predecessor modules that would set up the necessary conditions for them to be executed, so a chain of actions capable of transforming the current state to a goal state will be strongly activated by excitation flowing in both directions along the chain. Modules are organised into groups that compete for shared resources, and modules within a group inhibit each other. The most active module within each group, if its activation exceeds a threshold, is allowed to control the behaviour of the agent.

Like CS, ANA provides both compositional and implementational abstraction. In this case, compositional abstraction is achieved by allowing the *consummatory* action in a sequential chain (the action that finally achieves a goal set up by a chain of precursor actions) to represent the *goal* of that chain. Thus, the action "drink coffee" might be the consummatory action of a chain including "boil kettle," "pour water," "add coffee grounds," and "add cream." Activating "drink coffee" as a goal of the agent will tend to activate those precursor actions via a chain of connections. Implementational abstraction is possible in ANA by allowing more than one possible chain of precursor actions to converge on the same consummatory goal action. The state of the environment (via preconditions on individual actions) will then influence which of the action chains is actually effected.

There are clearly a number of similarities between the operation of ANA and CS. Both rely on activation spreading through a network of nodes. Both provide a form of compositional abstraction by allowing action sequences to be constructed from atomic actions. Both provide a form of implementational abstraction over multiple action sequences capable of achieving the same goal by allowing activation to spread from a goal to alternative actions that can achieve the goal. In both cases, the choice between alternative action sequences is made on the basis of activation level, and this choice can be influenced top-down by system goals, and bottom-up by the state of the environment.

There are, however, some important differences. The *schema* is not a fixed entity in ANA as it is in CS. A schema or program for action is constructed on line in response to a particular state of the world and a set of goals. Maes' approach emphasises automatic planning through chaining of modules via pre- and postconditions. In CS, by contrast, a schema is a fixed set of subgoals permanently represented within the CS hierarchy. A schema must exist for any sequence of actions that can be carried out by CS acting alone. Planning does not occur within CS but is the prerogative of SAS. SAS/CS thus explicitly locates planning within the deliberative layer of the architecture, as does Gat's three-layer formulation. The compositional abstraction of CS provides convenient clean primitives for planning to work within SAS, and the implementational abstraction provided by CS ensures that SAS plans can be carried out flexibly and adaptively. A consequence of the lack of a schema as a permanent abstraction over lower-level actions in Maes' approach is that it does not allow this hierarchical decomposition of complex action programmes.

One reason why the similarity between CS and ANA is interesting is that Maes' approach has been criticised as insufficient to provide reliable control of an agent carrying out realistic tasks. Tyrrell (1993) made three principal criticisms based on an implementation of ANA in an animal-like agent in a simulated environment.

First, Tyrell found problems with the way activation feeds along connections. In some cases, activation must be divided among recipient nodes; in others, it is fed equally to all of them. It is not possible to decide locally which to do in a particular case; it depends on whether the sources of activation are attempting to achieve the same or different goals (Tyrell, 1993, pp. 151–155). This arbitrarily biases, and thus disrupts, action selection.

Second, consummatory nodes (which finally achieve a goal set up by a chain of precursor modules) tend to lose out to modules earlier in competing chains (which merely prepare the way for some other goal to be achieved), all other things being equal. This is due to the module in the latter case having an extra source of excitation—unachieved modules later in the chain—not available in the former case. This leads to nonoptimal action selections in complex environments.

Third, ANA does not allow "compromise candidates"—actions that contribute to the achievement of more than one goal—to be promoted in the action selection mechanism.

While Maes reported that her mechanism works for limited planning problems, Tyrell found that it does not work well for the more general type of action selection problems studied by him.

The Norman and Shallice CS approach has been computationally implemented and applied to relatively complex real-world tasks in limited domains (Cooper & Shallice, 2000). It shows no evidence of the problems highlighted by Tyrrell with ANA. This is most likely because CS complies with three recommendations Tyrell makes regarding the ANA approach.

Tyrell (1993, pp. 153–154) identified the primary source of the first problem outlined above as the fact that, in ANA, excitation is fed into a chain of actions primarily by activating its consummatory node (representing its goal), which excites predecessor nodes. Tyrell recommended the alternative approach of exciting all the nodes in a behavioural sequence in parallel via a single higher-level (more abstract) representation of the sequence. This is, in fact, the approach taken in CS.

The second of Tyrell's criticisms is essentially avoided in CS by the same means. Activation is supplied to all child nodes of a schema in parallel and is not fed from node to node in a chain, allowing excitation of later nodes to decrease as earlier ones are executed and removed from the chain. Additionally, CS incorporates what Tyrell terms "contiguity." This means that there is an inbuilt bias toward continuing a sequence of actions within a particular schema once it has started, rather than starting a new competing schema, due to the internal dynamics of Cooper and Shallice's (2000) formulation, where each schema node feeds excitation back to itself as well as inhibits its competitors. Highly active schemas tend to persist until their child actions have been performed.

Finally, with respect to compromise candidates, Tyrell recommended hierarchical organisation of action nodes rather than the flat action chains of ANA. CS adopts an explicitly hierarchical organisation, allowing noncompeting branches of the schema hierarchy to be active in parallel.

The Maes and Tyrell models are couched in terms of the problem of action selection. The Norman and Shallice framework, however, identifying this level of control with its middle-layer control system, views such systems as providing abstraction over lower-level, less structured behaviour. We turn now to the higher-level system within the Norman and Shallice framework, which makes use of this abstraction layer to provide coarser-grained but more flexible control over behaviour.

Norman and Shallice Framework Compared with the Domino Agent Model

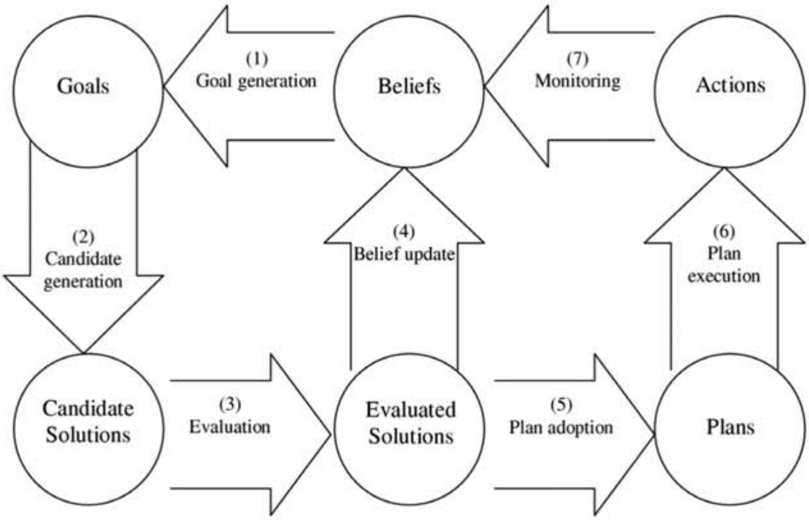
The top-level element in the Norman and Shallice framework, the SAS, is the locus of higher-level cognitive functions, including goal setting, planning, problem solving, monitoring of action, and detection of failure to achieve goals. The structure and operation of the SAS was initially somewhat poorly defined. Shallice and Burgess (1996) outlined the processes involved in the SAS and their relationships, based largely on neuropsychological evidence. However, the picture remained unclear, with many processes underspecified. This is largely due to the difficulty of obtaining clear empirical data on high-level psychological processes that are relatively distant from the sensory and motor periphery. However, while the SAS is a construct posed at a higher cognitive level than is typical for psychological theory, it addresses processes at the same general level as many theories in AI. This may allow psychological theory to benefit from the alternative perspective of AI, with an emphasis on engineering intelligent systems from first principles. Recently, progress has been made in specifying some aspects of the operation of the SAS, and a preliminary computational implementation has been produced (Glasspool, 2000; Glasspool & Cooper, 2002; Shallice, 2002). This has been done by making an analogy between the SAS and an established model of executive function in AI, the *Domino* framework of Das, Fox, Elsdon, and Hammond (1997).

Shallice and Burgess (1996) identified three stages in the operation of the SAS in its typical role of reacting to an unanticipated situation:

1. **The construction of a temporary new schema.** This is held to involve a problem orientation phase during which goals are set, followed by the generation of a candidate schema for achieving these goals.
2. **The implementation of the temporary schema.** This requires sequential activation of existing schemas in CS corresponding to its component actions.
3. **The monitoring of schema execution.** Because the situation and the temporary schema are both novel, processing must be monitored to ensure that the schema is effective.

The *Domino* model of Fox and colleagues (Figure 3) provides a framework for processes of goal setting, problem solving, and plan execution, and for understanding how they interact. It was arrived at based on experience developing a number of clinical decision support systems (Fox & Das, 2000). The nodes of the Domino can be viewed as state variables, while the arrows are inference functions that derive data of the type at the head of the arrow based on data of the type at the tail together with general knowledge and domain-dependent knowledge.

Figure 3. Domino agent framework of Das, Fox, Elsdon and Hammond (1997)



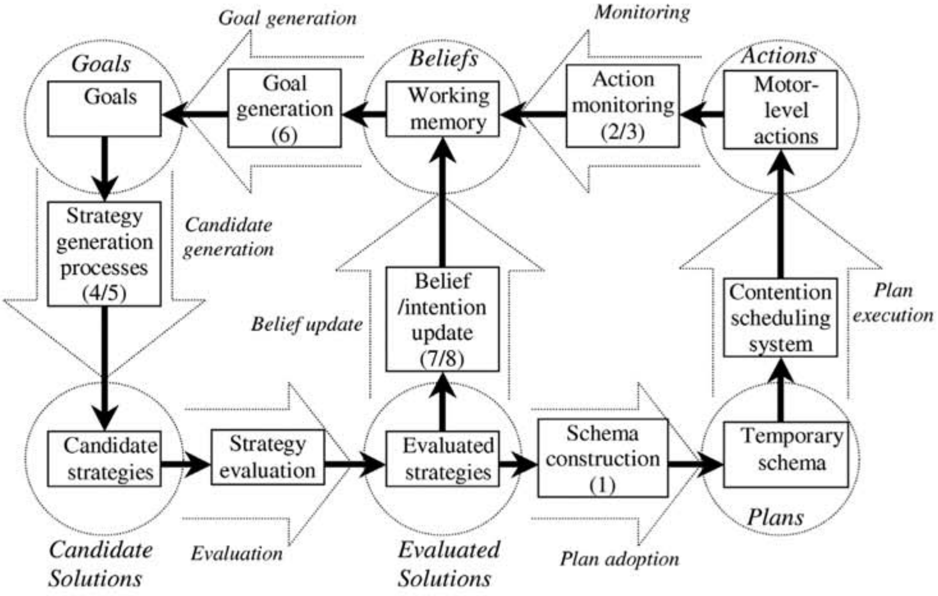
The framework specifies seven types of process operating on six types of information. In response to its current set of beliefs, the first process (1) raises goals for the agent. These form the input to Process (2), which applies problem-solving techniques to establish one or more candidate solutions, either potential new beliefs or potential plans of action, that could achieve the goal. Alternative solutions are evaluated (Process 3) to determine which should be adopted. The framework includes a proposed approach to evaluation based on logical argumentation (Fox & Das, 2000). Arguments for and against each candidate are established, and a decision is made based on the overall balance of argument, allowing one candidate to be adopted. If the candidate is a new belief, it is added to the belief database (4). If it is a proposed plan of action, it is adopted as the agent's current plan (5). The model extends to cover the implementation phase, in which actions of the plan are scheduled to meet various constraints on time, data, and other resources. Plans are decomposed into individual actions to be taken in the world (6), and the effects monitored (7), leading to new beliefs about the world. This may result in additional goals being raised (for example, to manage unexpected side effects or failures), which will initiate further cycles of decision making and planning.

The processes are similar to those specified by Shallice and Burgess: goal setting, solution generation and evaluation, decision making, planning, acting, and monitoring the effects of action. Glasspool (2000) and Glasspool and Cooper (2002) showed that the most important processes in Shallice and Burgess' outline diagram can be mapped directly onto the Domino framework. An advantage of doing this is that a set of well-understood and well-specified formal semantics can be associated with the framework (Fox & Das, 2000). This gives some reassurance that a model built on the framework will be computationally tractable. Figure 4 shows the mapping. The "candidate solution generation" process of the Domino framework corresponds to the generation of a strategy in SAS—a generalised plan of action that is subsequently implemented as a concrete schema for execution by the CS system.

The model of Glasspool (2000) is applied to a particular task—the Wisconsin card-sorting test (Milner, 1963; Nelson, 1976). Glasspool and Cooper (2002) added a second task, the six-element test (Shallice & Burgess, 1991). Both are tasks used clinically to assess frontal lobe function, and they are thus intended to specifically tax the types of executive function proposed to be located in the SAS.

A subject in the Wisconsin card-sorting test (WCST) is given cards that vary in the number, shape, and colour of the symbols they show. The subject is asked to sort the cards into piles, but they are not told the criterion for sorting. They might sort cards by the number of symbols, their colour, or their shape. After each card is placed, the experimenter indicates whether it was correctly sorted.

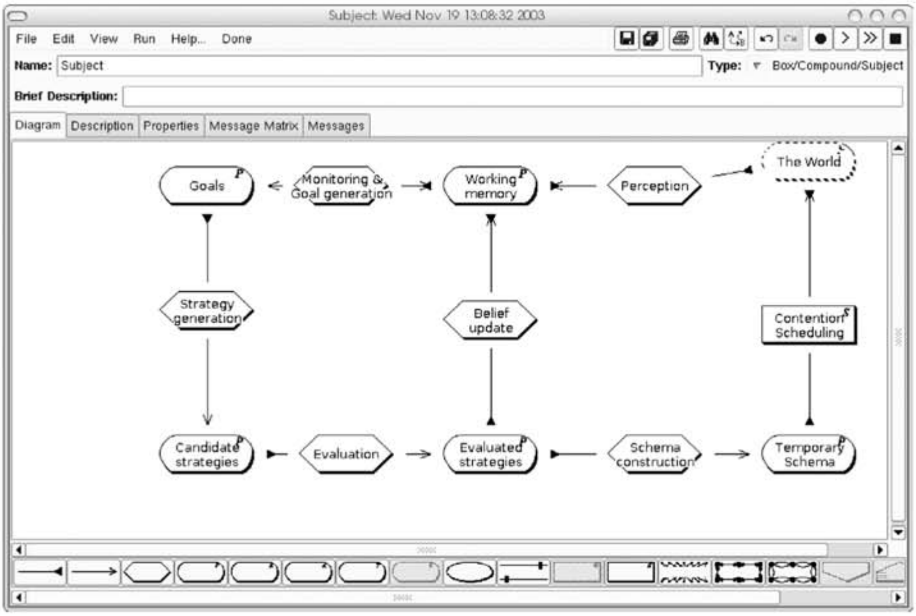
Figure 4. SAS outline of Shallice and Burgess (1996) mapped on to the Domino framework of Das et al. (1997) (broken lines) (Numbers in brackets refer to processes identified by Shallice and Burgess.)



Once the subject figures out the sorting criterion the experimenter is using, the experimenter changes to another sorting criterion without warning. "Normal" individuals typically catch on to the procedure quickly and make few errors. The six-element test (6ET) requires subjects to perform six simple tasks in 15 minutes. The test is constructed so that, while the tasks are simple, they cannot all be completed within the allotted time. Moreover, earlier responses within each task are scored higher than later responses, so the optimal strategy is to complete a little of each task, rather than completing any one task. Both tasks are designed to place demands on a number of aspects of executive processing, including the inhibition of prepotent responses, task switching, problem solving, monitoring progress toward current goals, and identifying errors and failures to achieve intended effects of actions. Patients with frontal lobe damage make many errors on these tasks.

The implementation of Glasspool (2000) and Glasspool and Cooper (2002) (Figure 5) is intended to be a first step toward an outline computational model of the SAS. It follows the Domino framework closely and assumes that contention scheduling fits into the *active* side of the framework, in the transition from plans to atomic actions in the world.

Figure 5. Outline implementation of the Shallice and Burgess SAS in the COGENT modelling system (Cooper & Fox, 1998) (Rounded boxes are buffers, square boxes are processes. The world is an external representation of the agent's environment. COGENT allows the boxes in the diagram to be fleshed out with computational specifications so that the model may be executed.)



Subject: Wed Nov 19 13:08:32 2003

Name: Subject

Type: Box/Compound/Subject

Brief Description:

The implementation assumes that SAS is involved in the initial generation of a strategy and the configuration of CS, which then attempts to carry out that strategy until it is interrupted by the SAS. In the WCST, the strategy is a card-sorting procedure, and the SAS interrupts to take back control when feedback indicates that the current sorting strategy is incorrect. In the 6ET, the strategy represents a decision to work on a particular task, and CS pursues the selected task until interrupted by SAS. The interrupt from SAS in this case is initiated by a temporal marker (Shallice & Burgess, 1996), indicating that a preset period of time has elapsed. SAS sets an appropriate marker when work is commenced on each task and switches to another uncompleted task each time a marker expires.

Shallice and Burgess suggested a number of procedures for strategy generation in response to a goal, the simplest of which is "spontaneous schema generation," the propensity of a suitable strategy to simply come to mind in response to a simple problem. In the current implementation, a process of this type is simulated by straightforward rules in "strategy generation." The appropriate rule for the

WCST may be paraphrased as: "If the goal is to sort an item into a category, and the item has distinguishable features, the item may be sorted according to one of those features." For the 6ET, the rule is as follows: "When given a list of tasks, work on one task for a period of time and then switch tasks."

The "strategy evaluation" process of Figure 5 ranks strategies basically according to two rules: Strategies that have recently been attempted and strategies that have recently proved unsuccessful are ranked lower than less recently attempted and less recently unsuccessful strategies.

This simple framework allows both the WCST and the 6ET to be simulated (Glasspool & Cooper, 2002). However, while this is an important first step, as yet, we have an incomplete understanding of the internal operation of these processes. Two important examples are the processes that determine appropriate goals given a particular set of beliefs and list appropriate candidate solutions that must be decided between. For limited domains, it is straightforward to implement these, but general solutions are more difficult to specify. Shallice and Burgess (1996) considered these issues in more detail, and a number of theoretical accounts exist in the AI and neuropsychological literature. Reconciling these with the Domino approach is an important area for further work.

The present implementation remains highly provisional. Little direct psychological evidence exists so far that can constrain the structures and processes posited even at this level of detail. The value of such an approach is that it provides a basis with which to begin building a detailed enough theory so that falsifiable predictions can be made. A wider benefit of a SAS simulation is the possibility of investigating the interface between SAS and CS. Learning is one important target for investigation. The CS system is held to acquire new schemas as a result of repeated application of the same strategy by SAS in similar situations. Once a schema has been acquired, the SAS is able to delegate operation to it without having to explicitly control behaviour. A number of processes are implicated in this SAS-to-CS transfer that cannot be studied without adequate characterisations of the two systems.

Another aspect of the interaction between SAS and CS is the need to remove the temporary schema (and possibly also deselect CS schemas) in response to novelty. It proved necessary to implement a special-purpose process within the model to quickly stop the CS system from carrying out whatever schemas are currently active when anomalous input is received from the environment. This was done in order to give the (significantly slower) SAS processes time to analyse and respond to the problem. Interestingly, such behaviour is also found in some robot control systems where a sufficiently powerful top-level executive system is present. For example, an autonomous spacecraft control system demonstrated by NASA (Muscettola et al., 1998) includes a process that puts the spacecraft into a standby mode (suspending routine operations) when an anomalous event occurs. Operation resumes when the anomaly has been analysed by executive systems and a new plan of action has been generated to deal with it. The need to add this behaviour to the model illustrates the advantage of simulation in the analysis of large-scale agent models. The interactions of multiple systems controlling behaviour with each other, with the agent as a whole, and with its environment can be difficult to analyse in the abstract.

Discussion and Conclusion

As an interdisciplinary field, cognitive science brings the rather different perspectives of cognitive psychology and AI to bear on the shared problem of understanding intelligent systems. The assumption driving cognitive science is that each field can provide insights to illuminate issues within the other. This chapter has highlighted some areas where cross-disciplinary insights may be drawn around a particular cognitive framework.

The Norman and Shallice framework can be viewed as a three-layer agent control architecture, and it is comparable with the class of three-layer architectures typified by Gat's work. The same roles for the three layers can be discerned in both cases. In this chapter, I argued that productive parallels can be drawn between the AI and psychological perspectives on this shared architecture. To conclude, some connections to related approaches in the literature will be outlined, and future prospects for the cross-disciplinary approach proposed in this chapter will be discussed.

The idea of layered control architectures is not new. Other layered schemes have emerged that identify somewhat different functions for the principal layers. An example is that of Sloman and colleagues, for example (Sloman, 2002), which has a layered arrangement with some additional mechanisms to control sensory input, motor output, and information flow between layers. The main difference from the style of the three-layer architecture described above is in the provision of sensory and motor mechanisms shared between layers. The lowest layer operates on reactive principles but includes both simple reflexive behaviour mechanisms and higher-level schemas that organise these into more complex sequences of action, providing abstraction. The second level provides deliberative processes, such as planning and reasoning about Level 1 processes, while processes at the third (meta-management) level reason about Level 2 processes. Thus, processes at Level 3 may monitor Level 2 processes looking for failures or loops in reasoning and planning. Such processes may provide error recovery and flexibility in choosing and changing strategy depending on the success of previous actions.

It is instructive to align the layers of Sloman's architecture (COGAFF) with those of the Gat-style TLA. The deliberative layer of the COGAFF architecture accords fairly well with Gat's deliberative layer. COGAFF includes a higher level, reflective layer that represents higher-order processes than are generally discussed in relation to the TLA; however, these would be seen as belonging to the TLA's deliberative layer. The COGAFF architecture thus splits the TLA deliberative layer into two distinct layers. Similarly, below the deliberative layer, COGAFF includes a single reactive layer, where the TLA has both abstraction and reactive layers. The functions of both of these layers are subsumed in the definition of COGAFF's lowest level, so here the TLA view splits a single COGAFF layer into two distinct layers. The primary difference as far as layering is concerned appears to lie in the location of layer boundaries, and this may reflect a difference in research priority between the two approaches —Gat's work focussing on the lower levels due to the need to produce working robots, and Sloman's taking more interest in higher-order functions. However, the inclusion of a TLA-style abstraction layer in a number of successfully implemented robots points to the utility of the concept.

A number of architectures for action selection have developed Maes' ANA approach in similar ways to those discussed above in relation to the CS architecture, in order to overcome the limitations pointed out by Tyrell. Bryson (2000) reviewed some of these. In Tyrell's terms, ANA and CS are action selection systems. This is not how CS is typically viewed from a psychological perspective, where the most salient feature of the system is the storage of precompiled schemas or motor programs. However, there are some interesting overlaps between the two perspectives. Action selection is effectively an action sequencing problem, where the driver for a sequence of actions is the state of the environment. Closely related models of endogenously driven sequential action (i.e., where the sequence of actions is driven by internal processes) have been developed in psychology (for example, see Glasspool, in press).

The various elements of the Norman and Shallice framework are now sufficiently well specified to allow computational implementation, and this probably constitutes the most interesting direction for future development of the approach outlined in this chapter. A comprehensive set of implementations of CS has been carried out by Cooper and Shallice (1997, 2000).

One interesting possibility suggested by the analogy between CS and the middle TLA layer is that CS might prove to be a good basis for the middle layer of a robot or software agent controller. The comparison with Maes' ANA suggests that CS might have a number of advantages in this role, and the task would be interestingly different from the neuropsychologically motivated tasks for which CS has thus far been simulated (Cooper & Shallice, 1997, 2000).

Specification of the SAS lags considerably; the present characterisation is little more than an outline of the processes involved. However, by taking advantage of frameworks developed from first principles within AI, such as the Domino agent framework, it is possible to build outline models that can act as the basis for further theoretical and empirical investigation. An important prospect offered by even an outline SAS implementation is the possibility to investigate interaction between the SAS and CS subsystems.

There are several aspects to this interaction. For example, what does it mean to say that SAS generates and implements a temporary schema to guide behaviour in a novel situation? What is required of the systems that monitor behaviour? How can they be integrated with SAS and CS? The initial SAS implementation has already hinted that a new process is required as part of this integration, to deselect CS schemas and, hence, halt habitual behaviour when unexpected feedback is received. Another important area is the acquisition of new schemas. SAS is held to control behaviour in situations where no appropriate schema exists, but if such a situation is encountered repeatedly, a new schema should be acquired within CS. This process might be rapid, generating a new schema in a single attempt, or it might be slow, the schema developing over time and gradually taking over more control from SAS, until CS is able to control the required behaviour alone. The relationship between CS and the competitive queuing (CQ) sequence generation model (Glasspool, in press) is potentially important here—such models can potentially provide a more complete account of endogenously driven sequencing of behaviour within CS than the current approach of relying entirely on environmentally driven preconditions on schemas. CQ systems are also able to learn action sequences through training. This may offer an interesting perspective on how new schemas can be acquired by CS. Some initial work to prepare this groundwork has already been done (Cooper & Glasspool, 2001). In order to investigate these aspects of the interface between the SAS and CS, both systems must be well characterised, hence, the advantage of computational implementations of both.

Attention is also required at the lower levels of the architecture. The interface between the CS and motor control layers is not yet clearly defined. The lower layer might be effectively implemented entirely within CS as a set of primitive schemas, or it might be a completely separate layer of control. The TLA perspective implies that the latter approach will be required if the system is to operate effectively, and the empirical arguments proposed by Cooper and Shallice (2000) tend to back up this view. More detailed computational modelling may provide a more principled boundary between these layers.

As well as being a promising basis for a large-scale model of cognition, the Norman and Shallice framework presents an interesting example of apparent theoretical convergence between AI and empirical psychology, and of the way in which theoretical work in both fields can benefit from interaction between them. There is significant potential within cognitive science for insight in one field to apply directly in another, especially given the different types of information used to motivate theories in AI and psychology. This is true at all levels of the cognitive system and is perhaps particularly important in attempting to develop fully integrated theories of cognition as a whole.

It is, of course, important to guard against the dangers of finding parallels where none exist, or of overgeneralising theories to make them fit to the extent that much that is important is lost. Overall, however, it is encouraging that such parallels can be drawn between the programmes of cognitive psychology and AI. A dialogue between AI and neuroscience on the problem of the control and integration of behaviour should benefit both fields. Approaches from AI and robotics may shed light on the structure of obscure higher processes in psychology. In turn, the increasingly detailed picture of human executive function emerging from neuropsychology can provide a rich context for theories of behaviour integration and control in AI.

Acknowledgments

I am most grateful to Richard Cooper, John Fox, Tim Shallice, and Heather Rhodes for numerous discussions on the material in this chapter.

References

Anderson, J. R., & Lebiere, C. (1998). *The atomic components of thought*. Mahwah, NJ: Lawrence Erlbaum Associates.

Bonasso, R. P. (1991). Integrating reaction plans and layered competences through synchronous control. In *Proceedings of the 12th International Joint Conference on Artificial Intelligence (IJCAI)*. Sydney, Australia: Morgan Kaufman.

Bonasso, R. P., Kortenkamp, D., Miller, D. P., & Slack, M. (1997). Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence, 9*.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence, 47*, 139–160.

Bryson, J. (2000). Cross-paradigm analysis of autonomous agent architecture. *Journal of Experimental and Theoretical Artificial Intelligence, 12*(2), 165–189.

Connell, J. H. (1991). A hybrid architecture applied to robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 2719–2724).

Cooper, R., & Fox, J. (1998). COGENT: A visual design environment for cognitive modeling. *Behaviour Research Methods, Instruments Computers, 30*, 553–564.

Cooper, R., & Glasspool, D. W. (2001). Learning action affordances and action schemas. In R. M. French & J. Sougne (Eds.), *Connectionist models of learning, development, and evolution* (pp. 133–142). London: Springer-Verlag.

Cooper, R. P., & Shallice, T. (1995). Soar and the case for Unified Theories of Cognition. *Cognition, 55*, 115–149.

Cooper, R. P., & Shallice, T. (1997). Modelling the selection of routine action: Exploring the criticality of parameter values. In M. G. Shafto & P. Langley (Eds.), *Proceedings of the 19th Annual Conference of the Cognitive Science Society* (pp. 131–136). Palo Alto, CA.

Cooper, R. P., & Shallice, T. (2000). Contention Scheduling and the control of routine activities. *Cognitive Neuropsychology, 17*, 297–338.

Das, S. K., Fox, J., Elsdon, D., & Hammond, P. (1997). A flexible architecture for autonomous agents. *Journal of Experimental and Theoretical Artificial Intelligence, 9*, 407–440.

Elsaesser C., & Slack M. G. (1994). Integrating deliberative planning in a robot architecture. In *Proceedings of the AIAA/NASA Conference on Intelligent Robots in Field, Factory, Service, and Space (CIRFFSS '94)* (pp. 782–787). Houston, TX.

Estes, W. K. (1972). An associative basis for coding and organization in memory. In A. W. Melton & E. Martin (Eds.), *Coding processes in human memory* (pp. 161–190). Washington, DC: V. H. Winston and Sons.

Firby, R. J. (1994). Task networks for controlling continuous processes. In *Proceedings of the Second International Conference on AI Planning Systems* (pp. 49–54). Washington, DC: AAAI Press.

Fox, J., & Das, S. (2000). *Safe and sound: Artificial intelligence in hazardous applications*. Cambridge, MA: MIT Press.

Gat, E. (1991). *Reliable, goal-directed reactive control of autonomous mobile robots*. PhD dissertation, Virginia Polytechnic Institute.

Gat, E. (1998). On three layer architectures. In D. Kortenkamp, R. P. Bonnasso, & R. Murphey (Eds.), *Artificial intelligence and mobile robots*. Washington, DC: AAAI Press.

Glasspool, D. W. (1998). *Modelling serial order in behaviour: Studies of spelling*. Doctoral thesis, University College London.

Glasspool, D. W. (2000). The integration and control of behaviour: Insights from neuroscience and AI. (Paper in symposium "*How to design a functional mind" at the AISB Convention, 2000.*) Technical Report 360, Advanced Computation Lab, Cancer Research UK.

Glasspool, D. W. (In press). Modelling serial order in behaviour: Evidence from performance slips. In G. Houghton (Ed.), *Connectionist modelling in psychology*. Hove, UK: Psychology Press.

Glasspool, D. W., & Cooper, R. (2002). Executive processes. In R. Cooper (Ed.), *Modelling high level cognitive processes* (pp. 313–362). Mahweh, NJ: Lawrence Erlbaum Associates.

Hartley, R., & Pipitone, F. (1991). Experiments with the subsumption architecture. In *Proceedings of the IEEE International Conference on Robotics and Automation* (pp. 1652–1658).

Hexmoor, H., & Kortenkamp, D. (1995). Issues on building software for hardware agents. *Knowledge Engineering Review, 10*(3), 301–304.

Houghton, G. (1990). The problem of serial order: A neural network model of sequence learning and recall. In R. Dale, C. Mellish, & M. Zock (Eds.), *Current research in natural language generation* (pp. 287–319). London: Academic Press.

Humphreys, G. W., & Forde, E. M. E. (1998). Disordered action schema and action disorganisation syndrome. *Cognitive Neuropsychology, 15*, 771–811.

Kaelbling, L. P. (1990). An architecture for intelligent reactive systems. In J. Allen, J. Hendler, & A. Tate (Eds.), *Readings in planning* (pp. 713–728). San Mateo, CA: Morgan Kaufmann.

Lhermitte, F. (1983). Utilisation behaviour and its relation to lesions of the frontal lobes. *Brain, 106*, 237–255.

Maes, P. (1991). The agent network architecture (ANA). *SIGART Bulletin, 2*(4), 115–120.

Milner, B. (1963). Effects of different brain lesions on card sorting. *Archives of Neurology, 9*, 990–1000.

Minsky, M. L. (1986). *Society of mind*. New York: Simon & Schuster.

Moravec, H. P. (1982). The Stanford Cart and the CMU Rover. *Proceedings of the IEEE, 71*(7), 872–884.

Muscettola, N., Nayak, P., Pell, B., & Williams, B. (1998). Remote agent: To boldly go where no AI system has gone before. *Artificial Intelligence, 103*(1–2), 5–48.

Nelson, H. E. (1976). A modified card sorting test sensitive to frontal lobe defects. *Cortex, 12*(4), 313–324.

Newell, A. (1990). *Unified theories of cognition*. Cambridge MA: Harvard University Press.

Nilsson, N. J. (1984). Shakey the robot. *SRI AI Center technical note 323*.

Norman, D. A., & Shallice, T. (1986). Attention to action: Willed and automatic control of behaviour. In R. J. Davidson, G. E. Schwartz, & D. Shapiro (Eds.), *Consciousness and self-regulation* (Vol. 4, pp. 1–18). New York: Plenum Press.

Reason, J. T. (1984). Lapses of attention in everyday life. In W. Parasuraman & R. Davies (Eds.), *Varieties of attention* (pp. 515–549). New York: Academic Press.

Schwartz, M. F., Reed, E. S., Montgomery, M. W., Palmer, C., & Mayer, N. H. (1991). The quantitative description of action disorganisation after brain damage: A case study. *Cognitive Neuropsychology, 8*, 381–414.

Shallice, T. (2002). Fractionation of the supervisory system. In D. T. Stuss & R. T. Knight (Eds.), *Principles of frontal lobe function* (pp. 261–277). Oxford: Oxford University Press.

Shallice, T., & Burgess, P. (1991). Deficits in strategy application following frontal lobe lesions. *Brain, 114*, 727–741.

Shallice, T., & Burgess, P. (1996). The domain of supervisory processes and temporal organization of behaviour. *Philosophical Transactions of the Royal Society of London B, 351*, 1405–1412.

Sloman, A. (2002). Architecture-based conceptions of mind. In P. Gardenfors, K. Kijania-Placek, & J. Wolenski (Eds.), *In the scope of logic, methodology, and philosophy of science* (Vol. II, pp. 403–427). Dordrecht: Kluwer.

Tyrell, T. (1993). *Computational mechanism for action selection*. PhD thesis, University ofEdinburgh.

Chapter 10

The CHREST Architecture of Cognition:

Listening to Empirical Data

Fernand Gobet
Brunel University, UK

Peter C. R. Lane
University of Hertfordshire, UK

Abstract

This chapter provides an introduction to the CHREST architecture of cognition and shows how this architecture can help develop a full theory of mind. After describing the main components and mechanisms of the architecture, we discuss several domains where it has already been successfully applied, such as in the psychology of expert behaviour, the acquisition of language by children, and the learning of multiple representations in physics. We highlight the characteristics of CHREST that enable it to account for empirical data, including self-organisation, an emphasis on cognitive limitations, the presence of a perception-learning cycle, and the use of naturalistic data as input for learning. We argue that some of these characteristics can help shed light on the hard questions

facing theorists developing a full theory of mind, such as intuition, the acquisition and use of concepts, the link between cognition and emotions, and the role of embodiment.

Introduction

In the last decade, an unprecedented amount of research has attempted to uncover the secrets of the human (and animal) mind. A number of approaches have been used, spanning philosophy, psychology, neuroscience, and computer science. This combined effort of several disciplines and tens of thousands of scientists has produced a formidable amount of new data and theoretical ideas. However, beyond this success in collecting new information, it has been more difficult to develop theories putting together, and thus explaining, large amounts of data, while still offering precise and well-specified mechanisms.

An influential line of research to achieve this goal has been the development of computational architectures that closely simulate human behaviour in a variety of domains. Examples of this approach include ACT-R (Anderson & Lebière, 1998), SOAR (Newell, 1990), and EPAM (Feigenbaum & Simon, 1984). More recently, the computational architecture CHREST (Chunk Hierarchy and REtrieval STructures) (Gobet et al., 2001; Gobet & Simon, 2000; Lane, Cheng, & Gobet, 2000) has simulated data in a number of domains, including expert behaviour, verbal learning, first language acquisition, and implicit learning.

The strength of cognitive architectures is that their implementation as computer programs ensures a high degree of precision and offers a sufficiency proof that the mechanisms proposed can carry out the tasks under study. Closer comparison between the theory's predictions and actual behaviour, using measures such as eye movements, reaction times, and error patterns, also establish the extent to which the simulation carries out the tasks in agreement with the human data.

The aim of this chapter is to provide an introduction to CHREST, to illustrate the kind of data it has already successfully simulated, and to show what insight it offers on some of the difficult questions facing researchers studying the mind, such as intuition, consciousness, implicit learning, and the link between emotion and cognition. The general approach defended here is that, in order to develop a theory of the mind, one has to use empirical data to constrain the number of possible architectures (see also Newell, 1990). Conversely, the chosen architecture should make new empirical predictions that can be tested and that can be used to further develop it. As we shall see, CHREST has already made new predictions in the field of expert behaviour—predictions that were later supported by empirical data (Gobet & Simon, 2000; Gobet & Waters, 2003).

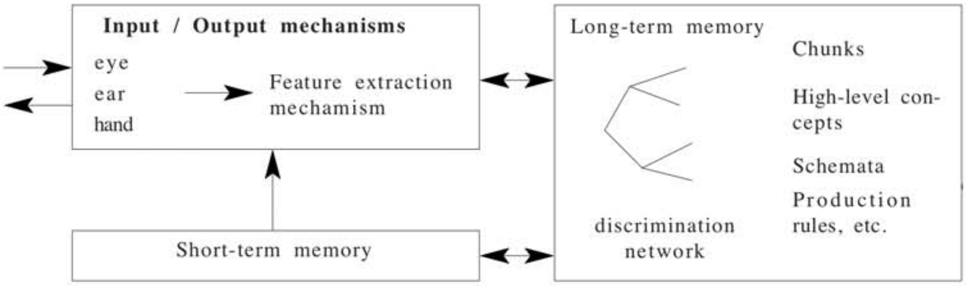
CHREST Architecture

In line with its predecessor, EPAM (Elementary Perceiver And Memorizer) (Feigenbaum & Simon, 1984), CHREST models the mind as a collection of emergent properties produced by the interaction of short-term memories (STMs), long-term memory (LTM), learning, perception, and decision-making structures and processes. In particular, it is assumed that there is a close interaction between perception, learning, and memory. For example, the system's knowledge will direct attention and perception, and, in turn, perception will direct the learning of new information. Another important characteristic of the architecture is that it is constrained by a number of parameters, such as the capacity of visual short-term memory, the rate at which new elements can be learned, and the time to transfer information from long-term memory to short-term memory. Thus, a theme central to this approach is that the human cognitive system satisfies Simon's requirements of bounded rationality (Simon, 1969). Finally, all operations carried out by the system have time costs, which make it possible to provide detailed simulations of empirical data. We now discuss CHREST's components and information-processing mechanisms in some detail.

Components

Figure 1 illustrates the main components of CHREST: (a) mechanisms and structures for interacting with the external world; (b) multiple STMs that hold information from varied input modalities; and (c) a long-term memory, where information is held in a "chunking network," which is a discrimination network that grows dynamically as a function of the inputs from the environment and the previous states of the system. Work on input–output channels, and the STMs associated with them, has focused on the role of visual attention, embodied by a simulated eye. As we will see later, this simulated eye plays an essential role in modelling expert perception and memory, and in understanding the interaction between low-level information, such as bitmaps, and high-level cognition, such as concepts. Some work has also been carried out on the detail of auditory inputs, with the best example being the study by Jones, Gobet, and Pine (2000), which simulated the way children learn words using phonemic information. Auditory STM was implemented using a combination of chunking mechanisms and a phonological loop (see also, Richman, Staszewski, & Simon, 1995; Zhang & Simon, 1985).

Figure 1. The CHREST computational model



Learning Mechanisms

The basic mechanisms for growing chunking networks are similar to those used in EPAM. They include mechanisms for creating new nodes and incrementally adding information to existing nodes. More recent research has examined mechanisms for creating high-level structures from perceptual information, such as schemata (known as "templates" within the CHREST community). The creation of schemata uses both stable information (for creating the core of a template) and variable information (for creating its slots). Templates are crucial for achieving realistic simulations of expert memory, in particular, to explain how chess masters can recall positions presented for just one or two seconds relatively well (Gobet & Simon, 2000). Another important extension is the automatic creation of lateral links (production links, similarity links, equivalence links, and generative links) between nodes in some circumstances (Gobet et al., 2001).

While CHREST maintains the emphasis shown by previous chunking models, for example (Simon & Gilmartin, 1973), on the number of chunks necessary to reach a high level of performance, it also stresses the necessity of building well-integrated and well-connected knowledge structures. Although this has often been discussed in the literature on expertise (Chi, Glaser, & Farr, 1988), CHREST's contribution is to provide computational mechanisms that automatically aggregate chunks into schemata and add links between existing chunks.

Time and Capacity Parameters

A key assumption in our approach is that cognitive architectures should be highly constrained by time and capacity parameters. CHREST includes approximate, but fixed, time parameters indicating the cost of carrying out cognitive processes. Memory parameters include the time to create a new node in LTM (8 seconds), the time to add information to an existing node by chunking (2 seconds), the time to store a chunk in STM (250 milliseconds), and the time to traverse a node during sorting through the chunking network (10 milliseconds) (De Groot & Gobet, 1996; Gobet & Simon, 2000). Further parameters apply to eye movements, such as the time to carry out a saccade (30 milliseconds). With respect to capacity, the main limitation is the span of short-term visual memory, which can hold only three chunks.

The emphasis on cognitive limitations contrasts with architectures such as SOAR, which have given precedence to the possibility of carrying out complex intelligent behaviour, while imposing relatively few constraints on the architecture (for example, the capacity of working memory is essentially unlimited in SOAR). Compared to other architectures, CHREST strikes the observer as an austere system. However, as a dynamic system, CHREST's behaviour is governed more by the complexities of its interaction with the environment than just its built-in capabilities. In common with chaos theory, the range of behaviours achieved by this deceptively simple cognitive model can come as a surprise.

Eye Movements and the Perception-Learning Cycle

A key question in cognitive science and artificial intelligence relates to the processes enabling a system to notice the relevant changes in the environment among the indefinite number of such changes. CHREST's answer to this question (sometimes known as the frame problem) has three parts. First, information from the environment is cut down by the limited capacity of the visual field; second, information is further limited by the (lack of) knowledge that the system brings to bear; and finally, it is further constrained by the memory capacity limits we have mentioned above. Obviously, if this description of intelligent cognitive systems is correct, these systems must have evolved powerful low-level mechanisms for extracting key features from the sensory input in order to survive complex and not necessarily friendly environments.

A further mechanism may be mentioned in relation to the frame problem. With CHREST, previous knowledge directs attention through eye movements, thus increasing the likelihood that critical information is heeded. This is based on the assumption that features that were critical in the past, and thus led to learning, are likely to be critical in the future. Furthermore, the focus of attention determines what will be learned. This perception-learning-perception cycle is another means of reducing the level of information extracted from the environment, so that responses may be made in real time.

Domains Already Successfully Addressed by CHREST

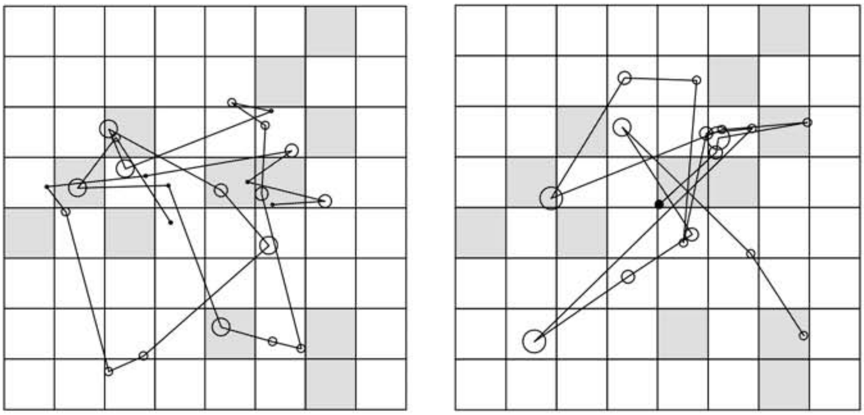
Chess Expertise

As mentioned above, we claim that a strong approach to developing a computational theory of the mind is to compare the predictions of the architecture with the empirical data. How does CHREST fare in this respect? CHREST was originally developed to explain expert behaviour in chess, a domain that draws upon a variety of cognitive abilities, such as perception, memory, decision making, and problem solving. CHREST is able to simulate a number of phenomena closely, such as novices' and chess Masters' eye movements (Figure 2), performance in memory-recall experiments (including errors and the detail of the piece placements), and how aspects of look-ahead search evolve as a function of skill (De Groot & Gobet, 1996; Gobet, 1997, 1998; Gobet & Simon, 2000). All of these phenomena are (partly) explained through the acquisition of a large number of chunks (more than 300,000 for Grandmasters), which are learned by scanning positions from databases of games taken from competitions.

A common criticism of computational modelling in general is that models only replicate behaviour (perhaps by means of some curve-fitting procedure) but do not lead to new predictions and thus do not produce real understanding. This criticism simply does not apply to CHREST, which has made a number of new predictions, some of which have been verified by empirical data.

The most striking example of a prediction made by CHREST is found in the recall of random chess positions, where the pieces of a game position are replaced haphazardly on the board. Most psychology textbooks report that Chase and Simon (1973) did not find any skill difference in the recall of such positions, while Masters perform much better than weaker players with game positions. This result has been taken as strong evidence that Masters use chunks of knowledge, as these are accessible in game but not in random positions. However, simulations with CHREST indicated that there should be a skill effect with random positions as well. The reason is that some chunks can be recognized in random positions by chance, in particular, with large networks of chunks. Reanalysis of

Figure 2. Example of a Master's eye movements for a specific position (left) and its simulations by CHREST (right) (The numbers below indicate the mean and the standard deviation of fixation times across all positions and all subjects. After De Groot & Gobet, 1996.)



Human Master **CHREST Master**

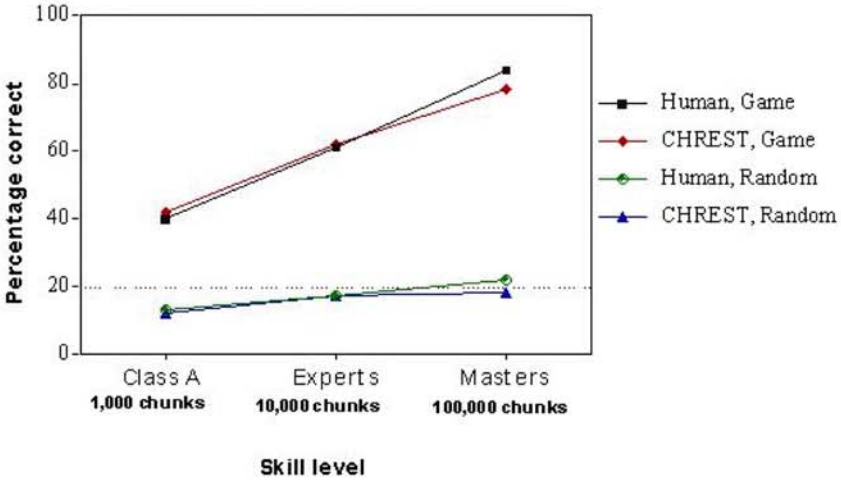
Human Master		CHREST Master
260 msec	*mean*	272 msec
100 msec	*sd*	97 msec

the literature, as well as new experiments, confirmed CHREST's prediction (Gobet & Simon, 1996, 2000) (Figure 3).

Vicente and Wang (1998) correctly noted that random positions, which are created by shuffling the locations of pieces in game positions, still contain information about the distribution of pieces in Master games. Stronger players may use this information in the memory task. For example, only one white and black king are allowed, not more than eight white pawns, and so on. Vicente and Wang proposed a new class of positions, which they called "truly-random positions," where both the locations and the distribution of pieces are randomized. Thus, a position could, in principle, contain 32 white pawns. For the same reason as with "classical" random positions, CHREST predicts a skill effect with truly random positions. By contrast, Vicente and Wang's theory, the "constraint-attunement hypothesis," based on the amount of goal-directed constraint available in the environment of a given task, predicts no skill effect.

In this instance, quantitative predictions were made before the data were collected. Gobet and Waters (2003) submitted players ranging from weak amateurs to strong Grandmasters to truly random positions, and they found a statistically reliable correlation between skill and recall performance. They considered several possible confounding variables and found that the correlation remained reliable even after variables such as age and visual memory were controlled for statistically.

Figure 3. Memory for game and random positions as a function of skill level, both for humans and CHREST (The human data are the average results aggregated across 13 studies. After Gobet & Simon, 1996.)



These results, which show that Masters can pick up patterns even in positions with little structure, suggest that chunking mechanisms operate automatically and implicitly. Unpublished work with Daniel Freudenthal indicated that chunking mechanisms can also explain phenomena in the implicit learning literature (Reber, 1967). In a typical implicit-learning experiment, participants see stimuli generated from an artificial grammar, and the question of interest is whether they can apply whatever they have learned (theories have focused around rules, fragments, and exemplars) to classify unseen strings as being well-formed or not. If the CHREST account is correct, participants in these experiments, albeit implicitly and unconsciously, are learning small fragments of stimuli that are incrementally augmented as learning progresses. (See Perruchet & Pacteau, 1990, for evidence supporting the role of fragments.)

Other Domains of Expertise

CHREST has also been applied to other domains of expertise, such as memory for computer programs and the acquisition of multiple representations in physics. The latter work is of particular interest in the context of this book. The aim was to investigate how novices studying a curriculum on electric circuits learn to combine two types of diagrammatic representation: the classic representation commonly found in physics textbooks, and a representation based on encoding quantitative properties of the domain in the diagrams (for details, see Lane et al., 2000). In modelling the acquisition of such knowledge, the simulated eye movements and the visual short-term memory play an essential role, both in learning chunks for the respective representations and in combining these representations. Perceptual chunks are also instrumental in learning to perform actions to draw the desired diagrams.

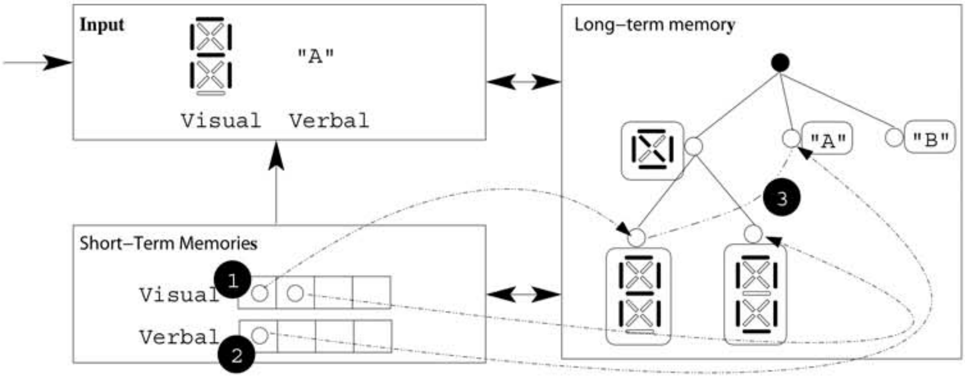
The work expands CHREST by adding some problem-solving abilities, which are admittedly still simple in comparison to those found in ACT-R or in SOAR. The emphasis in CHREST (or CHREST+, as we have called this problem-solving version) is on the role of perceptual chunks in planning the stages for constructing a diagram. Essentially, the chunks perceived in the problem are familiar, because they have been solved in isolation. Constructing a solution for the whole problem is guided by fitting together the solutions for the smaller chunks. The chunks used by CHREST in decomposing and solving diagrammatic problems correspond satisfactorily with those used and drawn by human participants; hence, we have empirical support for the role of perceptual chunks in problem solving.

Linking Perception to Expectations

An important characteristic of human, and perhaps animal, cognition is that information is often encoded in several modalities and with a fair amount of redundancy. Lane, Sykes, and Gobet (2003) explored how CHREST can encode information both visually and verbally, and how high-level knowledge can help disambiguate low-level information. The role of high-level knowledge is most clearly seen in the effect that prior expectations can have on the time and accuracy with which objects are recognized. The ultimate aim is to understand how low-level and high-level knowledge interact to produce intelligent behaviour, a question that has nagged cognitive science for decades (Neisser, 1966).

Figure 4 illustrates how CHREST learns an association between visual and verbal information. In the situation depicted, the model has been presented with a visual stimulus, and the verbal label "A." First, the visual stimulus is sorted through the discrimination network in long-term memory, and a pointer to the chunk retrieved is placed into the visual short-term memory. Second, the verbal stimulus is similarly sorted through long-term memory, and a pointer to the retrieved chunk is placed into the verbal short-term memory. Because the information in the different short-term memories is present simultaneously, the model learns a link between the two (marked 3 in the Figure): we call this link a *naming link*, because it can be used to name visual chunks.

Figure 4. Learning to link information across two modalities: (1) visual pattern is sorted through LTM, and a pointer to the node retrieved placed into visual STM; (2) verbal pattern is sorted through LTM, and a pointer to the node retrieved placed into verbal STM; and (3) a naming link is formed between the two nodes at the top of the STMs (After Lane, Sykes & Gobet, 2003.)



A limitation of CHREST is that input is provided in symbolic form. Current work aims to apply chunking mechanisms to bitmap-level data, while still using interactions between verbal and visuospatial information, on the one hand, and low-level and high-level knowledge, on the other. Extending CHREST in this manner emphasizes one of the strengths of its internal organisation of acquired chunks. Information in each modality is stored, compared, and matched in a manner relevant to that modality. Hence, bitmap images can be compared based on pixel-level grey-scale information. CHREST refers to particular instances of bitmaps by a pointer to the chunk containing that information. This symbolic reference supports links across modalities and the natural association of visual information with verbal descriptions.

Acquisition of Language

When discussing chess, we mentioned the link between expertise and implicit learning. Perhaps the standard example of such a link is the acquisition of language: most children become experts in their native language through the implicit learning of its constituents, including vocabulary and syntax. Although a small project simulated some data from the acquisition of vocabulary (Jones et al., 2000), our interest has centered on the acquisition of syntax (Croker, Pine,

& Gobet, 2001; Freudenthal, Pine, & Gobet, 2001; Gobet et al., 2001; Gobet & Pine, 1997). A variation of the CHREST architecture, known as MOSAIC (Model of Syntax Acquisition In Children), has successfully simulated several empirical phenomena in the early acquisition of syntactic categories. These include the "word island" phenomenon, which refers to the fact that syntactic proficiency is not acquired uniformly across an entire grammatical class, but is learned piecemeal, with some lexical items learned much faster than others from the same class (Pine, 1995; Tomasello, 2000); the "optional infinitive" phenomenon, which relates to the typical patterns of errors shown by children in their use of finite (for example, "goes," "went") and nonfinite verb forms (for example, "go," "going") (Wexler, 1998); and typical errors in the use of pronouns. The MOSAIC project is made unique within cognitive science by the combination of four features: the use of naturalistic input, taken from utterances spoken by mothers interacting with their children; the detailed simulations of patterns of errors and developmental trends at a lexical level; the fact that the same simulation can reproduce a variety of empirical phenomena; and the application of the model to several languages (so far, English, Dutch, and Spanish). Again, the extent of the coverage is surprising given that only simple and local learning mechanisms are used. MOSAIC's behaviour is the product of an interaction between rote learning, the creation and use of generative links, and the statistical structure of the input.

Another application of CHREST to developmental psychology may be mentioned here. Gobet (1999) reported simulations from the balance-beam task, a task originally developed by Inhelder and Piaget (1958), where children show a development characterized by stages. The aim was to propose a rough mapping between some key concepts of Piaget's (1970) theory of development (for example, assimilation and accommodation) with central mechanisms in EPAM/CHREST (for example, familiarization and discrimination). An interesting outcome of this research was that the model, while capturing some aspects of the data, showed difficulties similar to those encountered by connectionist models (McClelland & Jenkins, 1991; Raijmakers, van Koten, & Molenaar, 1996) in accounting for the discontinuities shown by children. Perhaps the lesson to be drawn from these studies is that the similar failing of two simple learning models point toward a more complex explanation.

Verbal Learning

More recently, we directed our attention to some of the verbal-learning experiments that were successfully simulated by EPAM (Feigenbaum & Simon, 1984). In these experiments, participants have to learn a list of pairs of nonsense syllables (such as XIM–BOJ), repeatedly presented in the same order. While our first interest was to develop material for a tutorial on CHREST, which we intend to offer in the future at major cognitive-science conferences, it was reassuring to see that we were able to successfully reproduce phenomena such as the serial position effect (items at the beginning and end of the list are learned faster than items in the middle), the von Restorff effect (items that are perceptually salient, perhaps because they are written in a different color, are learned rapidly), and the importance of strategies in carrying out these tasks. Our implementation also suggested that the time parameters for learning interact in complex ways with the strategies used, a result that was not discussed by Feigenbaum and Simon.

Interim Summary

Although the research described above covers a broad range of domains, the small size of the CHREST community means that coverage is clearly less than that of architectures such as ACT-R or SOAR. It should be pointed out, however, that CHREST incorporates many fewer degrees of freedom than these theories. It also shows a higher degree of self-organisation than alternative architectures, where the hand coding of productions is required before simulations can be run. We now turn our attention to some projects we have just started pursuing, before discussing some phenomena we wish to tackle in the future.

Developing a Complete Mind: Some Hard Questions

While CHREST is far from implementing a complete mind, we can be confident that the components in place are at least sufficient to produce humanlike behaviour in a number of domains, sometimes with a remarkable degree of detail. In this section, we offer some speculations, with varying degrees of empirical support, on how CHREST can help explain some difficult questions in cognitive science.

Intuition and Insight

Intuition refers to the rapid understanding shown by experts facing a new problem. *Insight* refers to the sudden discovery of a solution after a lengthy and unsuccessful search. There is substantial empirical evidence supporting the presence of these two phenomena. For example, firefighter commanders rely on intuition to make a decision in situations characterized by high risk and high time pressure (Klein, 1998). Typically, they do not consider alternative choices but quickly adopt the appropriate behaviour. The same type of decision-making behaviour has been reported, among other domains, in intensive-care nursing, battle commanders, and rapid chess. Similarly, evidence for the phenomenon of insight, often referred to in scientists' biographies (Langley et al., 1989), includes well-controlled experiments (Kaplan & Simon, 1990).

It is generally accepted that understanding the human mind requires a detailed explanation of the phenomena of intuition and insight. Simon (1979) has forcefully argued that intuition can be explained by pattern-recognition mechanisms and insight by a combination of pattern-recognition, search, and learning. We agree with his explanations, but we also think that detailed simulations of these phenomena with a computational architecture such as CHREST would lend more strength to his conclusions. In particular, simulations could shed light on whether active constructive mechanisms are also required for intuition, and, if not, why this is not the case (see chapter 9 of De Groot & Gobet, 1996). Preliminary work along these lines was reported by Gobet and Jansen (1994), who showed that a variation of CHREST, called CHUMP (CHUnking of Moves and Patterns), was able to learn an association of chess moves to perceptual patterns. In the context of this discussion, it is of particular interest that CHUMP, with an admittedly relatively low skill level, did better in positions requiring a "positional judgement" than in tactical positions, which engage more look-ahead search. Both in the popular and the technical literature (Dreyfus & Dreyfus, 1986), positional judgement in chess is seen as a paradigmatic example of intuition.

The Role of Concepts in a Mind

The formation and use of concepts for categorization is essential if organisms are to interact effectively with their environment. Without categories, each instance would have to be treated in isolation, and learning would show little generalization. There is a substantial experimental literature about how people create and use concepts, and several computational models have been developed to account for these results. In particular, Gobet et al. (1997) used a variation of EPAM to simulate the role of strategies in learning Brunswick faces (Medin & Smith, 1981), a task that is typical of experiments carried out in this field. The current version of CHREST also sheds new light on how concepts may be organized in humans. We argue that what counts as a "concept" is not a single chunk, or even a single template, in the discrimination network, but a subset of nodes linked by a varying density of lateral links—concepts are thus represented in a distributed fashion. While similar ideas have been proposed in the literature, the contribution of CHREST is to provide mechanisms explaining how the nodes and the links between them are created automatically as the program receives input from the environment. In addition, the work described in Lane, Sykes, and Gobet (2003) is important for exploring mechanisms explaining how conceptual information relating to different modalities is integrated.

What is lacking, however, is a detailed theory of how concepts are used when organisms are situated in their environments, for example, in decision-making or social situations. For example, how do categories mediate between perception and action in complex problem-solving situations? We believe that CHREST is also uniquely positioned for studying these questions, as the learning processes leading to the creation of chunks and templates, as well as the addition of lateral links, offer plausible mechanisms to explain how concepts are acquired. More importantly, the capability of the architecture to seamlessly link perception and action makes CHREST an ideal platform for exploring how concepts are used in problem-solving situations. To some extent, work on the role of templates in chess and on the acquisition of multiple representations in physics has already started to address these questions in simulation experiments. The work on the acquisition of language, in particular on how syntactic structures are learned, can also be seen as an attempt to explore how simple mechanisms operating on naturalistic inputs lead to the creation of categories.

How Emotions and Motivation Relate to Cognition

Minds have emotions and motivations, and researchers have explored means of incorporating them within computational architectures (Belavkin, Ritter, & Elliman, 1999; Fellous, Armony, & LeDoux, 2002; Simon, 1967; Sloman & Logan, 1999). In a classical paper, Simon (1967) argued that with systems characterized by serial organisation and control hierarchy, motivation refers to what is controlling attention at a specific time; in particular, given that these systems have multiple goals, motivation controls how attention is focused on a specific goal. Furthermore, it is necessary to have a provision for interrupt mechanisms; Simon proposes that at least two sources inform these mechanisms: first, drives (for example, hunger), and, second, the information gained by EPAM's process of noticing.

Recent work with Philippe Chassy aims to add (at least some) emotions to CHREST. We are exploring several possibilities. First, emotions mediate some of the parameters of the architecture, such as the time to create new long-term memory structures or to store information into STM. Second, emotional tags are added to chunks and templates during the learning process. For example, knowledge that a physics problem is difficult may link a given chunk to physiological mechanisms associated with negative affects. Finally, emotions may be linked to the equilibrium, or lack thereof, between discrimination and familiarization processes, perhaps as a way to interrupt the unfolding of either of these processes. We are currently testing these speculations by closely comparing CHREST behaviour with that of humans under different experimental conditions.

Future Trends

Embodiment

Embodied cognition has been an active domain of research in recent years, and several mobile robots have been developed that carry out simple tasks (Pfeifer & Scheier, 1999). We have argued elsewhere (Lane & Gobet, 2001) that current research fails to link simple behaviour with more complex behaviours, and that systems showing intelligence require both levels of complexity. In particular, the failure of current embodied cognition research to incorporate mechanisms dealing with symbolic processing seriously limits the types of behaviours that can be accounted for.

The question of symbol grounding has a simple explanation within the CHREST framework: chunks (which are symbols) are grounded through perception. Critically, as noted above, chunks also affect the way the world is perceived, and this is done actively by directing eye movements. We agree that these arguments would be more forceful if we could demonstrate an implemented system embodying these ideas. We are currently working to implement CHREST into a mobile robot. Our first aim is to show that a chunking-based architecture can replicate some of the "classic" simulations in the literature, such as a mobile robot's ability to avoid obstacles. Our belief is that combining symbolic with nonsymbolic approaches within an embodied system is likely to have important consequences, both for theory and application. In particular, symbolic information will enable the robot to be "articulate," explaining, verbally, what and why it makes particular decisions.

Consciousness

The importance of unconscious and implicit cognition was already apparent when we discussed expertise and the acquisition of language. This raises the questions of how and why human minds are conscious—the holy grail of cognitive science. Roughly speaking, this question can be divided into two parts: how minds pay attention to and show awareness of objects in their environment, and how minds reach the subjective experience of existing. The line of research starting with EPAM and now being continued by CHREST has a lot to say about the former (Ericsson & Simon, 1993; Richman, Gobet, Staszewski, & Simon, 1996; Simon, 1997). A strong, testable prediction is that organisms are conscious only of the information stored in STM, and that they are not conscious of the information and processes used during learning and recognition. Thus, the capacity of STM plays an important role in limiting the number of objects of which we are conscious. The work on intuition, on creating automatisms through chunking, and on the role of pattern recognition in expertise is of direct relevance here.

By contrast, EPAM and CHREST, and cognitive science in general, have much less to contribute to the second component of the question—the phenomenological experience of "I-ness." Nonetheless, these architectures may be used to derive some principles underpinning consciousness. We speculate that the architectural limits inherent in CHREST, as well as its ability to implicitly acquire aspects of language, may be used to address this question. In particular, language enables the model to refer to both verbal and nonverbal information, including a notion of "self." Of course, how this notion comes about is a tough question. We suggest that the necessary conditions for its unfolding include a close link between perception, learning, and action; the presence of a physical body, including emotions, so that what is being learned from the environment is linked to stable spatial and affective references; STMs sufficiently large to enable recursive construction of knowledge (this implies that some species may not reach consciousness, because this capacity is too limited); and finally—and this is missing in most research on embodied cognition—mechanisms for creating symbols and relating them to nonsymbolic knowledge. In order to flesh out these speculations, an architecture such as CHREST may be used to make computational explorations, which may, in turn, lead to designing experiments with humans or animals.

Conclusion

A reasonable criticism of our approach is that it mainly uses symbolic (although domain-representative) input to train the architecture. As mentioned earlier, we are currently starting to use less abstract types of input, such as bitmap images and sensor input from a mobile robot. While we have focused upon the psychological aspects of CHREST, it must be realized that there are also potential technical applications in the area of artificial intelligence. For example, the architecture's ability to seamlessly combine low-level and high-level aspects of cognition can be used in image analysis.

Although we are obviously biased toward our approach, we also believe that true progress in understanding the human mind will be made by comparing cognitive architectures, both those closely simulating empirical data and those developed based more on basic architectural principles, such as the necessity of parallelism or the need to minimize computation. An interesting question is whether the conclusions reached by simulating the detail of human behaviour will coincide with conclusions based upon architectural considerations.

The achievements of CHREST in explaining a variety of empirical phenomena entitle us to suggest some ideas about what a complete model of the mind would look like. Interaction between perception and cognition, the capability to incrementally acquire a vast and integrated structure of nodes and links, as well as (paradoxically) strong architectural limits are the key features behind CHREST's success in simulating these data, and, we suggest, will also be behind any successful computational theory of mind. While carrying out detailed simulations of human behaviour is time-consuming and difficult, it may be the most efficient way to search the space of possible architectures for a mind.

References

Anderson, J. R., & Lebière, C. (Eds.). (1998). *The atomic components of thought*. Mahwah, NJ: Erlbaum.

Belavkin, R. V., Ritter, F. E., & Elliman, D. G. (1999). Towards including simple emotions in a cognitive architecture in order to fit children's behaviour better. In M. Hahn & S. C. Stoness (Eds.), *Proceedings of the 21^{st} Conference of the Cognitive Science Society* (p. 784). Mahwah, NJ: Erlbaum.

Chase, W. G., & Simon, H. A. (1973). Perception in chess. *Cognitive Psychology, 4*, 55–81.

Chi, M. T. H., Glaser, R., & Farr, M. J. (Eds.). (1988). *The nature of expertise*. Hillsdale, NJ: Erlbaum.

Croker, S., Pine, J. M., & Gobet, F. (2001). Modelling children's case-marking errors with MOSAIC. In E. M. Altmann, A. Cleeremans, C. D. Schunn, & W. D. Gray (Eds.), *Proceedings of the Fourth International Conference on Cognitive Modeling* (pp. 55–60). Mahwah, NJ: Erlbaum.

De Groot, A. D., & Gobet, F. (1996). *Perception and memory in chess: Heuristics of the professional eye*. Assen: Van Gorcum.

Dreyfus, H., & Dreyfus, S. (1986). *Mind over machine*. New York: Free Press.

Ericsson, K. A., & Simon, H. A. (1993). *Protocol analysis: Verbal reports as data* (2nd ed.). Cambridge, MA: MIT Press.

Feigenbaum, E. A., & Simon, H. A. (1984). EPAM-like models of recognition and learning. *Cognitive Science, 8*, 305–336.

Fellous, J. M., Armony, J., & LeDoux, J. E. (2002). Emotion and computational neuroscience. In M. A. Arbib, *The handbook of brain theory and neural networks* (2nd ed.). Boston, MA: MIT Press.

Freudenthal, D., Pine, J. M., & Gobet, F. (2001). Modelling the optional infinitive stage in MOSAIC: A generalisation to Dutch. In E. M. Altmann, A. Cleeremans, C. D. Schunn, & W. D. Gray (Eds.), *Proceedings of the Fourth International Conference on Cognitive Modeling* (pp. 79–84). Mahwah, NJ: Erlbaum.

Gobet, F. (1997). A pattern-recognition theory of search in expert problem solving. *Thinking and Reasoning, 3*, 291–313.

Gobet, F. (1998). Expert memory: A comparison of four theories. *Cognition, 66*, 115–152.

Gobet, F. (1999). Simulations of stagewise development with a symbolic architecture. In J. P. Dauwalder & W. Tschacher (Eds.), *Dynamics, synergetics and autonomous agents* (pp. 143–156). Singapore: World Scientific.

Gobet, F., & Jansen, P. (1994). Towards a chess program based on a model of human memory. In H. J. van den Herik, I. S. Herschberg, & J. E. Uiterwijk (Eds.), *Advances in Computer Chess 7* (pp. 35–60). Maastricht: University of Limburg Press.

Gobet, F., Lane, P. C. R., Croker, S., Cheng, P. C. -H., Jones, G., Oliver, I., & Pine, J. M. (2001). Chunking mechanisms in human learning. *Trends in Cognitive Sciences, 5*, 236–243.

Gobet, F., & Pine, J. M. (1997). Modelling the acquisition of syntactic categories. In *Proceedings of the 19th Annual Meeting of the Cognitive Science Society* (pp. 265–270). Hillsdale, NJ: Erlbaum.

Gobet, F., Richman, H., Staszewski, J., & Simon, H. A. (1997). Goals, representations, and strategies in a concept attainment task: The EPAM model. *The Psychology of Learning and Motivation, 37*, 265–290.

Gobet, F., & Simon, H. A. (1996). Recall of rapidly presented random chess positions is a function of skill. *Psychonomic Bulletin and Review, 3*, 159–163.

Gobet, F., & Simon, H. A. (2000). Five seconds or sixty? Presentation time in expert memory. *Cognitive Science, 24*, 651–682.

Gobet, F., & Waters, A. J. (2003). The role of constraints in expert memory. *Journal of Experimental Psychology: Learning, Memory and Cognition, 29,* 1082–1094.

Inhelder, B., & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence*. New York: Basic Books.

Jones, G., Gobet, F., & Pine, J. M. (2000). Learning novel sound patterns. In N. Taatgen & J. Aasman (Eds.), *Proceedings of the Third International Conference on Cognitive Modelling* (pp. 169–176). Veenendaal, The Netherlands: Universal Press.

Kaplan, C. A., & Simon, H. A. (1990). In search of insight. *Cognitive Psychology, 22*, 374–419.

Klein, G. A. (1998). *Sources of power: How people make decisions*. Cambridge, MA: MIT Press.

Lane, P. C. R., Cheng, P. C. -H., & Gobet, F. (2000). CHREST+: Investigating how humans learn to solve problems using diagrams. *AISB Quarterly, 103*, 24–30.

Lane, P. C. R., & Gobet, F. (2001). Simple environments fail as illustrations of intelligence: A review of R. Pfeifer & C. Scheier: "Understanding Intelligence." *Artificial Intelligence, 127*, 261–267.

Lane, P. C. R., Sykes, A. K., & Gobet, F. (2003). Combining low-level perception with expectations in CHREST. In F. Schmalhofer, R. M. Young, & G. Katz (Eds.), *Proceedings of EuroCogSci 03: The European Cognitive Science Conference 2003* (pp. 205–210). Mahwah, NJ: Erlbaum.

Langley, P., Simon, H. A., Bradshaw, G. L., & Zytkow, J. M. (1987). *Scientific discovery*. Cambridge, MA: MIT press.

McClelland, J. L., & Jenkins, E. (1991). Nature, nurture and connections: Implications of connectionist models for cognitive development. In K. VanLehn (Ed.), *Architectures for intelligence*. Hillsdale, NJ: Erlbaum.

Medin, D. L., & Smith, E. E. (1981). Strategies and classification learning. *Journal of Experimental Psychology: Human Learning and Memory, 7*, 241–253.

Neisser, U. (1966). *Cognitive psychology*. New York: Appleton-Century-Crofts.

Newell, A. (1990). *Unified theories of cognition*. Cambridge, MA: Harvard University Press.

Perruchet, P., & Pacteau, C. (1990). Synthetic grammar learning: Implicit rule abstraction or explicit fragmentary knowledge? *Journal of Experimental Psychology: General, 199*, 264–275.

Pfeifer, R., & Scheier, C. (1999). *Understanding intelligence*. Cambridge, MA: MIT Press.

Piaget, J. (1970). *L'épistémologie génétique*. Paris: PUF.

Pine, J. (1995). First verbs and what they tell us. *First Language, 15*, 77–101.

Raijmakers, M., van Koten, S., & Molenaar, P. (1996). On the validity of simulating stagewise development by means of PDP networks: Application of catastrophe analysis and an experimental test of rule-like network performance. *Cognitive Science, 20*, 101–136.

Reber, A. S. (1967). Implicit learning of artificial grammars. *Journal of Verbal Learning and Verbal Behaviour, 6*, 855–863.

Richman, H. B., Gobet, F., Staszewski, J. J., & Simon, H. A. (1996). Perceptual and memory processes in the acquisition of expert performance: The EPAM model. In K. A. Ericsson (Ed.), *The road to excellence*. Mahwah, NJ: Erlbaum.

Richman, H. B., Staszewski, J. J., & Simon, H. A. (1995). Simulation of expert memory with EPAM IV. *Psychological Review, 102*, 305–330.

Simon, H. A. (1967). Motivational and emotional controls of cognition. *Psychological Review, 74*, 29–39.

Simon, H. A. (1969). *The sciences of the artificial*. Cambridge, MA: MIT Press.

Simon, H. A. (1979). *Models of thought* (Vol. 1). New Haven, CT: Yale University Press.

Simon, H. A. (1997). Scientific approaches to the question of consciousness. In J. D. Cohen & J. W. Schooler (Eds.), *Scientific approaches to consciousness*. Mahwah, NJ: Erlbaum.

Simon, H. A., & Gilmartin, K. J. (1973). A simulation of memory for chess positions. *Cognitive Psychology, 5*, 29–46.

Sloman, A., & Logan, B. (1999). Building cognitively rich agents using the Sim_agent toolkit. *Communications of the ACM, 42*, 71–77.

Tomasello, M. (2000). The item-based nature of children's early syntactic development. *Trends in Cognitive Sciences, 4*, 156–163.

Vicente, K. J., & Wang, J. H. (1998). An ecological theory of expertise effects in memory recall. *Psychological Review, 105*, 33–57.

Wexler, K. (1998). Very early parameter setting and the unique checking constraint: A new explanation of the optional infinitive stage. *Lingua, 106*, 23–79.

Zhang, G., & Simon, H. A. (1985). STM capacity for Chinese words and idioms: Chunking and acoustical loop hypothesis. *Memory and Cognition, 13*, 193–201.

Chapter 11

Managing Goals and Resources in Dynamic Environments

Elizabeth Gordon
University of Nottingham, UK

Brian Logan
University of Nottingham, UK

Abstract

A key problem for agents is responding in a timely and appropriate way to multiple, often conflicting goals in a complex, dynamic environment. In this chapter, we propose a novel goal-processing architecture that allows an agent to arbitrate between multiple conflicting goals. Building on the teleo-reactive programming framework originally developed in robotics, we introduce the notion of a resource *that represents a condition that must be true for the safe concurrent execution of a durative action. We briefly outline a goal arbitration architecture for teleo-reactive programs with resources that allow an agent to respond flexibly to multiple competing goals with conflicting resource requirements.*

Introduction

One of the defining characteristics of an autonomous agent is its ability to generate its own goals in response to changes in its environment. At any given time, such an agent will typically have several goals; for example, a package delivery robot may have a goal to deliver a package to a particular office (triggered by a user request), a goal to keep its battery charged (innate), and a goal to avoid colliding with the person who has just stepped out of their office (autonomously generated). A key problem with goal-based architectures is *goal arbitration*, i.e., which goal or goals to work on next. Ideally, the agent should work to achieve as many goals as possible, while respecting any priority ordering over goals and the limitations imposed by its environment and effectors. The agent should be able to respond to opportunities and threats as they arise, while continuing to work toward its existing goals (to the extent to which this is possible). However, many existing goal-based agent architectures only allow an agent to work toward a single goal at a time. In this chapter, we extend the teleo-reactive framework described by Benson and Nilsson (1995) to allow an agent to work toward multiple goals simultaneously. We introduce the notion of a resource representing a condition that must be true for the safe concurrent execution of a teleo-reactive program, and we present an algorithm for goal arbitration between teleo-reactive programs with resources.

The examples in this chapter and the implementation we describe involve agents that act as characters in a computer game. The domain of computer games is becoming increasingly popular as a research platform for artificial intelligence (Laird & Duchi, 2000; DePristo & Zubek, 2001; Hawes, 2000). Developing an agent for a computer game is essentially the same as any simulation-based approach to artificial intelligence (AI). Modern computer games are essentially simulated worlds. While games are simpler than the real world, they provide a range of locations, situations, objects, characters, and actions that present game characters with a complex, dynamic environment. Most computer games are real-time, and the environment can be changed by a human player or other characters. Games, therefore, present a challenging agent design problem, forcing us to confront issues of real-time action selection in pursuit of multiple goals.

Although our approach has been developed in the context of games, it is general and can be applied to any complex dynamic environment or resource-dependent task. It is situated within a body of work that takes agent architectures as central, for example, Bryson's work on modular agent designs (2001) and Wright's work on cognitive architectures for emotional agents (1997), both of which are discussed.

The remainder of the chapter is organised as follows. In the next section, we present a brief introduction to the teleo-reactive framework as described in Nilsson (1994), Benson and Nilsson (1995), and Benson (1996) and highlight some of the problems that form the motivation for our work. We then introduce the notion of a resource, which represents a condition that must be true for the safe concurrent execution of a durative action, and we sketch a new goal-processing architecture for agents, GRUE, that extends teleo-reactive programs with resources. We briefly describe the main components of the GRUE architecture and present an algorithm for goal arbitration using resources. We conclude with a brief discussion of related work and outline directions for future work.

Teleo-Reactive Programs

The teleo-reactive framework (Benson & Nilsson, 1995; Benson, 1996) was developed to control agents in dynamic environments and represents an attempt to blend ideas from control theory with standard computer science techniques in order to give the agent the continuous feedback necessary to operate effectively (Nilsson, 1994). In this section, we discuss the teleo-reactive framework. The next section discusses our approach, which builds on the teleo-reactive framework.

The teleo-reactive (TR) framework comprises a library of plans, an arbitrator, a planner, and a learning system. The plans in the TR framework are teleo-reactive programs. A teleo-reactive program (TRP) consists of a series of rules, each of which consists of a condition and an action. The program is run by evaluating all the rules and executing the first rule with a condition that evaluates to be true when matched against a world model stored in the agent's memory. The conditions are evaluated continuously, ideally by an electric circuit, but otherwise, continuous evaluation is simulated by executing the smallest time steps practical for the application (Nilsson, 1994).[1] This allows the agent to respond quickly to changes in the environment. Actions that continue executing as long as a condition remains true are called *durative actions*. Correspondingly, we will refer to conditions that are evaluated continuously as *durative conditions*. When simulating continuous evaluation, a rule with a condition that evaluates to be true during a time-step either executes an action or starts a durative action. If the durative action has already been started, the rule simply causes the action to continue executing. Durative actions terminate when the rule that started them ceases to fire.

Each TRP achieves a single goal. The first rule in a TRP encodes the goal condition achieved by the program and performs the null action. The next rule contains an action that can make the goal condition true, and so on. Each action achieves a condition higher in the list of rules. This is referred to as the regression property (Nilsson, 1994). Plans are constructed at runtime by the planner or selected from a set of existing plans in the agent's plan library.

A TR agent attempts to achieve a set of goals that may be prespecified by the agent designer or provided by a human operator. The arbitrator allows a TR agent to work on several goals at once, by determining which plan should be allowed to perform an action at each cycle (Benson & Nilsson, 1995; Benson, 1996). Plans are chosen using the concept of stable nodes. A stable node is a point in a plan at which execution of the program can safely be suspended. That is, the work done up to that point in the plan is stable with respect to the other plans that are running. A condition is stable if running the other plans will not cause the condition to become false. So, for example, a plan used by a package delivery agent might require the agent to pick up an object and take it somewhere. The condition of having the object is stable with respect to any plan that does not require the agent to drop the object. The set of stable nodes is compiled before the plans start running using STRIPS-style delete lists, and it only needs to be recompiled when the set of plans changes.

During each execution cycle, the arbitrator runs the plan with the best expected reward/time ratio. The reward is the reward the agent expects for achieving the goal (which may or may not actually be received), and the time is the estimated time necessary to reach the closest stable node. The arbitrator uses stable nodes to avoid undoing things it has already done. Stable nodes are safe places to stop programs, so the arbitrator runs each program until it reaches one; it can then switch to another program if appropriate. This allows the agent to take small amounts of time to achieve less-rewarding goals, while it is also working on a more time-consuming but more rewarding goal. When a plan achieves its goal and runs the null action, it is removed from the arbitrator.

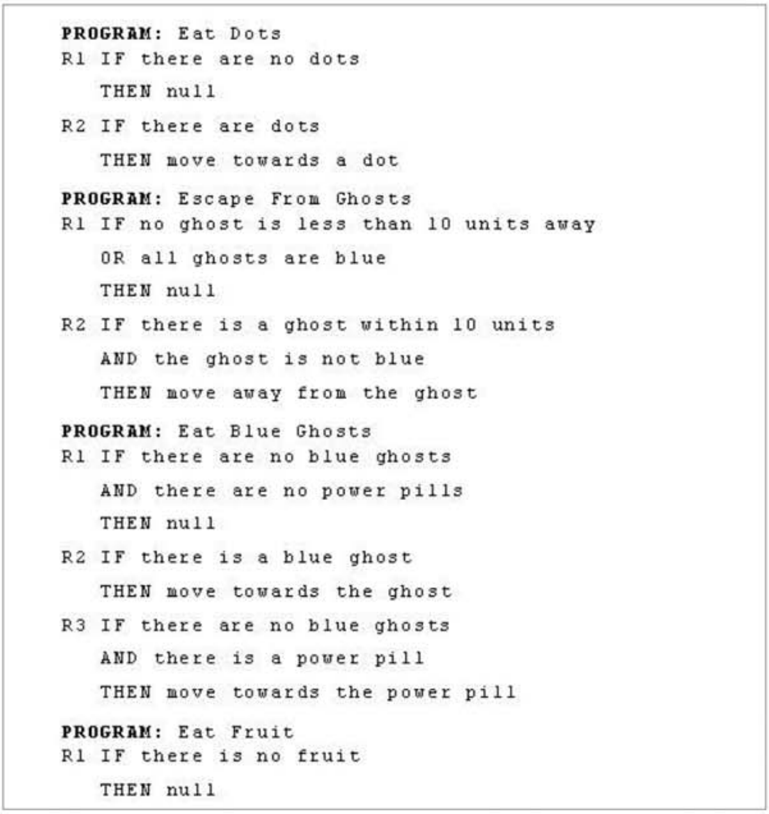
Example: Pacman

As an illustration of the TR framework, we present a collection of teleo-reactive programs that might be used to play the game Pacman. We remind the reader that in the game, the player controls a yellow character (Pacman) who moves around a maze eating dots in order to earn points. The maze contains hazards in the form of multicolored ghosts. The player starts with three lives, and the game is over when all the lives are gone. If a ghost catches Pacman, the player loses a life, and Pacman is placed in a safe position. There are also special dots called power pills that Pacman can eat to make the ghosts vulnerable (and blue colored). While the ghosts are blue, Pacman can eat them and earn points. The player's score can also be increased by eating fruits, which sometimes appear in the middle of the maze. Note that Pacman is always eating—it is not possible for Pacman to move over an edible object without eating it. Our Pacman agent will play Pacman as if it were a human player. That is, it can see the entire maze and all the ghosts at any time.

We have chosen to implement the Pacman agent using four teleo-reactive programs that achieve four top-level goals of the game: eating dots; escaping from ghosts, which allows the player to stay alive and continue play; eating blue ghosts; and eating fruit. Pseudo-code for the four programs is given in Figure 1.

At any given moment, the Pacman agent is running one of the above programs, say the program for eating blue ghosts. The rules are examined from the top down, so if there are no blue ghosts and no power pills, then there is nothing to do. Let us assume there are power pills, so we continue to the next rule. Rule 2 states that if there is a blue ghost present, we should chase it. However, if there

Figure 1. Pseudocode for the Pacman game



```
PROGRAM: Eat Dots
R1 IF there are no dots
   THEN null
R2 IF there are dots
   THEN move towards a dot
PROGRAM: Escape From Ghosts
R1 IF no ghost is less than 10 units away
   OR all ghosts are blue
   THEN null
R2 IF there is a ghost within 10 units
   AND the ghost is not blue
   THEN move away from the ghost
PROGRAM: Eat Blue Ghosts
R1 IF there are no blue ghosts
   AND there are no power pills
   THEN null
R2 IF there is a blue ghost
   THEN move towards the ghost
R3 IF there are no blue ghosts
   AND there is a power pill
   THEN move towards the power pill
PROGRAM: Eat Fruit
R1 IF there is no fruit
   THEN null
```

are no blue ghosts, we continue to the third rule, which tells us to eat a power pill to turn the ghosts blue, thus enabling us to switch to using the second rule.

When processing the rules for the *Eat Blue Ghosts* program, the arbitrator checks to see if the highest condition that is currently true is stable with respect to other TR programs, and if so, switches to the program with the best reward/time ratio (which may be the *Eat Blue Ghosts* program). However, Pacman is a fast-paced game, making most conditions unstable. In the set of programs above, we can identify only a few conditions, such as "no fruit" being present, that will not be changed by any of the other programs. However, this is not very useful as a stable node, as "no fruit" being present is a success condition, so if it is true, the *Eat Fruit* program will stop running. Other conditions, such as "there is a fruit," are unstable, because any program that causes Pacman to move might cause Pacman to eat the fruit. This is due to the nature of the game—there is no eat command, Pacman simply eats anything it runs into.

Limitations of Teleo-Reactive Programs

Teleo-reactive programs have a number of advantages for controlling agents in dynamic environments like Pacman. They gracefully handle changes in the environment, whether those changes force the program to go back to previous steps or allow it to jump ahead. The arbitrator allows a teleo-reactive agent to perform actions that work toward multiple goals (Benson & Nilsson, 1995), in that the arbitration algorithm can switch to a different teleo-reactive program for each execution cycle, effectively running all the programs in pseudo-parallel.

However, the standard teleo-reactive architecture has a number of limitations. One problem is with the approach to goal arbitration, which relies on the notion of stable nodes. The presence or absence of stable nodes is determined by the set of actions possible in the environment. In some environments, it is difficult to find stable nodes, making it impossible to use them to allow pseudo-parallel execution of TRPs. The distribution of stable nodes throughout the active plans imposes a minimum latency on responses to changes in the environment or the agent, because the agent will only consider switching tasks when it reaches a stable node. In our Pacman programs, there were no stable nodes that were not success conditions. If we were to try to switch between these programs at stable nodes as in Benson and Nilsson (1995), the arbitrator would run only one program at a time. However, it is easy to see that running *Escape from Ghosts* can cause Pacman to move toward (and hence eat) a power pill, which also makes progress toward the goal of *Eat Blue Ghosts*, suggesting that another approach may be more effective in this environment.

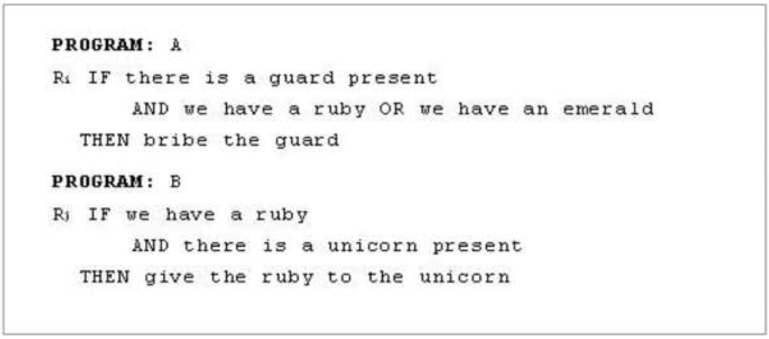
If there are few stable nodes, arbitration can have the effect of forcing the agent to work toward the unachieved goal with the highest reward to the exclusion of all other goals. As an example, we built a simple agent that plays the game Unreal Tournament. Unreal Tournament has several game modes; our agent plays one in which each player has a flag and can gain points by stealing their opponent's flag. The game is combat based, so players can use weapons to attack each other. The player's health is measured on a scale of 1 to 100, and there are health items that restore about 20 points. Our agent uses a basic strategy of always guarding its flag. It has goals for regaining health, attacking an opponent, and returning to its flag. Each goal has a corresponding teleo-reactive program, but only one program can run during each execution cycle. The goal for regaining health has the highest reward, which means that if the agent's health is low, only the program for regaining health will run. The problem is that once a health item has been used, it takes some time for another one to appear. The regain health goal is triggered whenever the agent's health is below a threshold. If the agent's health is so low that it is below the threshold even after using a health item, it will simply wait for the health item to reappear. In effect, it stands still in the middle of a room, making itself an easy target. This is not effective or realistic behaviour. More generally, if two programs are the same length and have the same expected reward, but one of the two has fewer stable nodes, then that program will have a lower reward/time ratio, and as a result, it will not be favored during arbitration.

Using stable nodes requires a list, for each action, of conditions that are falsified by that action. To compute the set of stable nodes, we therefore need to be able to predict the effects of actions in the environment. Because the effects of an action may depend on the current environment, this, in turn, requires that we have a complete domain model, or the agent programmer has the long and tedious task of listing every possible action, in every possible situation, with every possible consequence.

Even in those cases where a domain model is available, the need to compute stable nodes places restrictions on the syntax of rules. In particular, we cannot have disjunctive conditions in rules. For example, look at the two rules given in two different programs, as shown in Figure 2. Assuming that both a guard and a unicorn are present, running the rule in *Program B* will make the condition of that in *Program A* false *only if the agent does not have an emerald*. Whether or not the agent has an emerald cannot be determined until runtime.

More generally, the teleo-reactive architecture described in the literature (Benson & Nilsson, 1995; Benson, 1996) is not completely autonomous. Goals and the rewards for achieving them are either predefined by the developer or are given by the user. The former is inflexible, and the latter is often inappropriate; for example, in a computer game where agents must be truly autonomous, the human players should be playing the game, not providing input to their opponent.

Figure 2. Examples of programs with disjunctive rules



```
PROGRAM: A
Ri IF there is a guard present
       AND we have a ruby OR we have an emerald
  THEN bribe the guard
PROGRAM: B
Rj IF we have a ruby
       AND there is a unicorn present
  THEN give the ruby to the unicorn
```

GRUE: A New Architecture

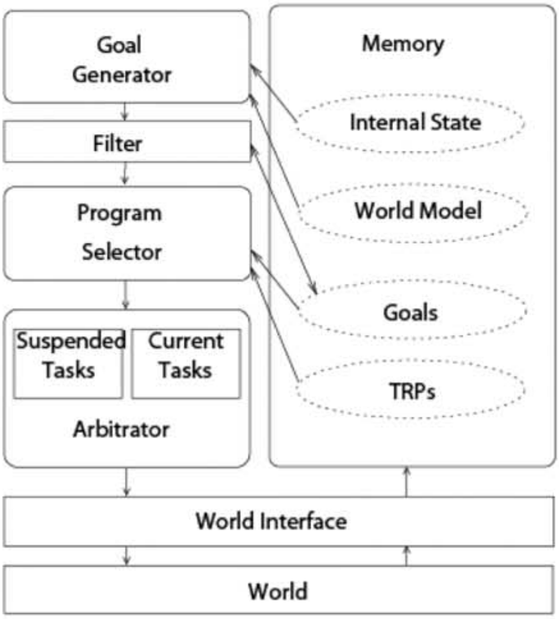
Therefore, we developed a new teleo-reactive architecture, GRUE (goal and resource-using architecture), that overcomes these limitations:

- It provides support for goal generators, allowing an agent to generate new top-level goals and adjust the priorities of existing goals in response to the current environment.
- It allows the agent to make progress toward multiple goals in environments with few stable nodes.
- It allows true parallelism rather than co-routining, with multiple programs running actions in parallel during each cycle where this is possible.

Our architecture contains four main components: a memory, a set of goal generators, a program selector, and the arbitrator (Figure 3). The system runs in cycles. During each cycle, information from the environment is processed by the world interface and then placed in memory. Goal generators are triggered by the information in memory, then associated with programs by the program selector, and then executed by the arbitrator.

The agent operates in a world that could be a game engine, the physical world (for a robot), or any other environment in which the agent operates. The world interface takes information from the environment, in this case, a game engine, does any necessary preprocessing, for example, the distance between game objects and the agent, and stores the results as resources in the memory module.

Figure 3. GRUE (Goal and Resource Using architecturE)



The information obtained through the world interface may include information about the agent, if it is obtained from the agent's sensors. Information may additionally be placed in memory by the agent's programs; the distinction is that information processed by the world interface is information about the agent's interaction with the world. Data produced by the agent's programs are purely the result of internal processing. The memory module stores all information in a single database. This allows for easy access to all types of information. The ellipses in Figure 3 indicate the different types of information and their uses but are not meant to suggest that they are stored separately. Goal generators are triggered by the presence of particular information in memory. They create appropriate goals, computing the priority values as necessary. For example, a goal generator might be triggered when the agent's health is below a particular value. It would then generate a goal to regain health, with a priority that is inversely proportional to the agent's current health value. The agent designer can choose not to run the goal generators at every cycle, trading reduced execution time for an increase in the time it takes for a character to respond to events in the environment.[2] Once created, goals pass through a filter that eliminates low-priority goals. Adjusting the filter threshold can adjust the number of goals entering the arbitrator. The program selector is a simple lookup function, which appends the appropriate teleo-reactive program to each goal to create a task. It is provided mainly to allow for possible future additions of planning or learning components. The arbitrator manages a list of current tasks and decides which task(s) to run at the current cycle. The overall aim of the arbitrator is to run as many tasks as possible, subject to resource constraints, and to give priority to tasks that achieve more important goals.

In the remainder of this section, we discuss the main data structures used by the program and their associated algorithms in more detail. We begin by outlining the contents of the agent's memory.

Resources

A key component of GRUE that distinguishes it from similar architectures is the way in which TRPs can obtain exclusive access to items in memory, indirectly precluding the execution of competing TRPs. This exclusive access is implemented using resources and resource variables.

Informally, a *resource* is anything necessary for a rule in a program to run successfully. Objects in the agent's environment are the most obvious example, but other more abstract things, like facts, properties of the agent, or time periods, can also be regarded as resources. More precisely, a resource consists of a unique identifier naming the resource together with a set of resource properties. Each property is an attribute value pair, consisting of a property name and a property value. The set of relevant properties will depend on the application but would typically include those features of the agent and its environment that are relevant to the agent achieving its goals.

Resources are output by the world interface (for example, the agent's sensors) and are stored in the agent's memory. Each resource is represented as a list consisting of an identifier (a string) followed by one or more property-name-property-value pairs. Property names are labels (strings), and property values are constants (strings or numbers). For clarity, we write property names in uppercase. For example, a health item might be represented by the structure

```
(HealthPack101 [TYPE HealthPack] [HEALTH-PTS 20])
```

which indicates that the item `HealthPack101` is of type `HealthPack`, and will restore 20 health points to the agent. Properties can be multivalued and divisible. A *multivalued* property can have more than one value for the same resource.[3] For example, more than one category may be listed for the `TYPE` field, for example, a pickaxe might be represented as

```
(Pickaxe102 [TYPE pickaxe] [TYPE weapon])
```

which indicates that the resource Pickaxe102 is a pickaxe and also a weapon.

In general, a resource can only be used by a single task at a time. However, a property that is declared to be *divisible* can be split, with part being used for one task and part being used for another. This is often convenient when tasks require a number of identical resources, such as moments of time, rounds of ammunition, or units of money. For example, we could represent 10 coins as 10 different resources or as a single resource with a divisible property AMOUNT. Divisible resources are treated specially by the binding algorithm (see following). If a numeric property value is prefixed by the DIV keyword, the binding algorithm treats the property value as a divisible quantity. For example, the resource

```
(Gold101 [TYPE money] [AMOUNT DIV 53])
```

can be used to buy several things, as long as the total cost is less than or equal to 53. When a resource variable binds part of a resource, the remainder is put back into the list of available resources so it can be used by another program. Note that we must use the DIV keyword and then split the resource during the binding process, as until then, there is no way to tell how many portions the resource should be divided into.

Resource Variables

A *resource variable* represents a resource that must be available throughout the execution of a durative action. Resource variables are placeholders for resources in the condition of a rule in a TRP. Resource variables are matched against the resources in the agent's memory. Matching can be constrained so as to specify that only resources with particular properties are selected. When writing GRUE TRPs, we specify conditions in terms of properties of resources and require that rules only run when the resource variables in its condition can be bound.

A resource variable is a 4-tuple containing the following:

- An identifier for this variable
- A flag indicating whether the variable requires exclusive access to the resource

- A set of required properties
- A set of preferred properties

The identifier is the variable name (a string) that is bound to a resource. It can subsequently be used to access any of the properties of the resource. By convention, resource variable identifiers begin with a "?" character. The mutual exclusion flag determines whether resource variables in other programs can bind to a resource bound by this variable. If the mutual exclusion flag has the value "SHARED," then the resource can be bound by other programs. For example, a rule that simply checks the existence of a resource or that extracts information from a resource that represents information about the environment can share the resource with rules in other programs. However, in cases where a resource is deleted by a rule, or where a rule's actions require sole access to the resource, the flag should have the value "EXCLUSIVE." Note that mutual exclusion applies between programs—two exclusive variables in the same rule can bind to the same resource. Required properties are those that are necessary for the execution of a rule's actions or the maintenance of a goal condition (for maintenance goals). Preferred properties are used to choose between resources when more than one resource matches all the required properties. Programmers can use preferred properties to give agents different behaviors by specifying preferences for different kinds of otherwise equivalent resources. For example, a rule condition in an attack program might contain the following resource variable:

```
(?Gun104 EXCLUSIVE ([TYPE gun] [AMMUNITION 20]) ())
```

which specifies a weapon with 20 rounds of ammunition and no preferred properties. It also specifies that the rule requires EXCLUSIVE access to the resource. (For example, another program cannot simultaneously sell or give the gun to another agent while it is being used by the attack program.) If a particular kind of weapon is preferred, we can specify this using a preferred property:

```
(?Gun106 EXCLUSIVE
  ([TYPE Gun] [AMMUNITION 20])
  ([TYPE RocketLauncher]) ())
```

The following example matches a piece of information previously stored by a running program:

```
(?BaseLocation107 SHARED ([TYPE location]) ())
```

and it allows other programs to bind the resource.

Numerical values are handled slightly differently. It is often useful to be able to specify that the value of a particular property should lie within a particular range and, additionally, to specify an ordering over values within that range. For example, a program might require an object as close to the agent as possible (smallest distance value) or prefer a weapon with more ammunition. We use the following notation to specify such additional constraints on resource variable matching.

A *value range* consists of brackets containing two numbers representing the extent of the range and, optionally, a utility arrow. We define two types of brackets: "#|" and "|#" denote a range with firm lower and upper bounds, and the brackets "#:" and ":#" denote a range with soft lower and upper bounds. A firm bound indicates that a number must be within the range to be considered a match. A soft bound indicates that numbers within the range have the highest utility, but values outside the indicated range are still considered to match. All range boundaries are inclusive, so a firm lower bound of five would match five or more. For example, the following resource variable matches any monster between one and 10 units away, inclusive:

```
(?Monster1 EXCLUSIVE ([TYPE monster] [Distance #|1 10|#]) ())
```

The two types of brackets can be mixed to represent ranges, with one firm bound and one soft bound. When one bound of a range is soft, we allow the corresponding number to be omitted. If the soft bound is on the right, we assume that the range ends at +∞, and if it is on the left, at -∞. (We expect that in most cases, it will be clearer to specify the lower end of the range explicitly.) For example, the following resource variable asks for an amount of money with a value of at least 10 and no upper limit:

```
(?Gold102 EXCLUSIVE ([TYPE money] [Amount #|10 :# ]) ())
```

The arrows, → and ←, are used to indicate a utility ordering over matching values. So, if we want to specify that closer monsters are preferred, then we can add an arrow:

```
(?Monster1 EXCLUSIVE ([TYPE monster] [Distance #|1←10|#]) ())
```

Similarly, the following resource variable requires a minimum of 10 units of money but specifies that more is better:

```
(?Gold103 EXCLUSIVE ([TYPE money] [Amount #|10 → :# ]) () )
```

Figure 4 illustrates the relative utility of property values in the soft-bounded range #:5 –> 10:#. Contrast this to Figure 5, which shows a range with one firm bound and one soft bound.

Figure 6 specifies a minimum of five and an upper limit of +∞. Figure 7 specifies a maximum of five and a lower limit of -∞. Note that the utility ordering can go in either direction. The range #| 5 <– :# indicates a maximum utility at five, and a minimum utility at +∞.

Finally, * can be used as a special wildcard symbol. It is used in situations where we want to require that a resource has a particular property listed, but we do not care about the value. For example, the resource variable:

```
(?Box115 ([TYPE Container] [LOCATION *]) ())
```

allows us to ask for a container for which we know the location, without requiring a particular value for that location.

The resource variables in each rule in a program are matched against the resources stored in memory. A resource variable matches only those resources that have all of the properties listed in its required properties list. If there is more than one resource that matches the required properties, the resource variable matches the resource with the largest number of preferred properties. If two or

Figure 4. Value range #:5–>10:#

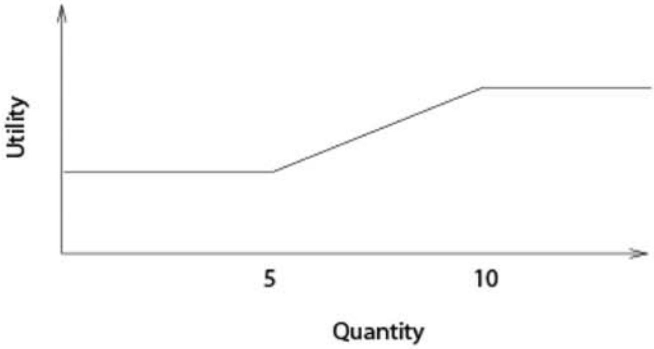
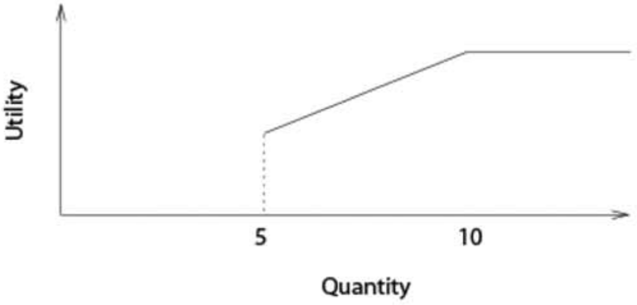


Figure 5. alue range #|5–>10:#



more resources have all the required properties and the same numbers of preferred properties, then one resource will be chosen arbitrarily.

If the resource variable specifies a property value range for a required or preferred property, binding works slightly differently. Required properties are checked as normal, with a numerical quantity matching a range if it is within the range. (For ranges with soft lower and upper bounds, any number matches.) For preferred properties, all resources with the specified required properties are checked to see whether the value of the preferred property lies within the range. Then, all the matching property values are evaluated to find the best match according to the utility ordering specified by the range. For example, if the range is #| 5 –> 10 |#, then a value of nine is better than a value of six. The resource with the highest utility is considered to match the preferred property, and all the other resources are then treated as if they did not match.

Once a variable has been bound to a resource, we can use the property function to retrieve additional information. If the resource variable:

```
(?Gun116 EXCLUSIVE ([TYPE gun]) ())
```

is bound to the resource

```
(Revolver116 [TYPE weapon] [TYPE gun] [AMMUNITION 20])
```

Figure 6. Value range #|5–>:#

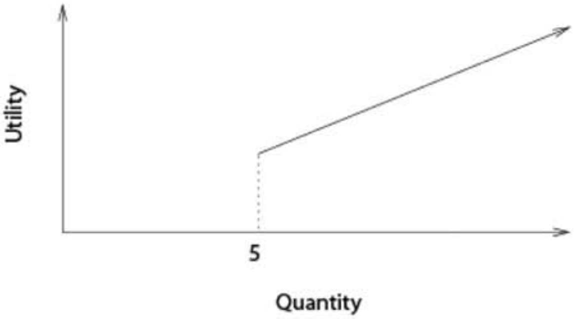
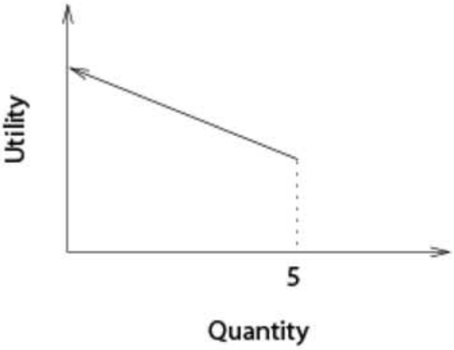


Figure 7: Value range #:<–5|#



we can then ask how much ammunition ?Gun116 has:

```
property(?Gun116, AMMUNITION) = [20]
```

Goals

A *goal* is a list consisting of an identifier (a string), a priority (an integer), a type (`ACHIEVEMENT` or `MAINTENANCE`), and the name of a program that will achieve or maintain the goal condition (a string).

Achievement goals are straightforward. For example, we can write the goals for our Pacman agent as follows:

```
(GetAwayFromGhostsGoal 90 ACHIEVEMENT EscapeFromGhosts)
(EatBlueGhostsGoal 50 ACHIEVEMENT EatGhosts)
(EatFruitGoal 50 ACHIEVEMENT EatFruit)
(EatDotsGoal 20 ACHIEVEMENT EatDots)
```

Maintenance goals, where the agent is attempting to maintain a condition, are a special case. At first glance, it seems that a teleo-reactive program that achieves a goal will maintain the goal state automatically—if a necessary condition stops being true, the program will automatically try to make it true again. However, teleo-reactive programs are normally removed from the arbitrator when their goal conditions are achieved. The goal condition must be violated for the goal to be regenerated and the corresponding maintenance task to reenter the arbitrator.[4] This can result in one or more goal conditions being achieved intermittently, rather than maintained. For example, a player may sell his or her weapon to get money to buy a potion, notice that he or she is without a weapon (a violation of a maintenance goal), and then buy it right back again.

Tasks achieving maintenance goals, therefore, persist in the arbitrator, even when the goal condition is (currently) achieved. Only the top rule in the TRP will fire, producing a null action, but the rule can bind those resources necessary to maintain the condition. Maintenance goals should have appropriate priority so they can be used to prevent the character from disposing of necessary items or "forgetting" to maintain a crucial condition.

The type field in the goal data structure is used to distinguish between maintenance goals and ordinary goals, for example:

```
(MaintainHealthPointsGoal 95 MAINTENANCE RegainHealthPoints)
```

Programs

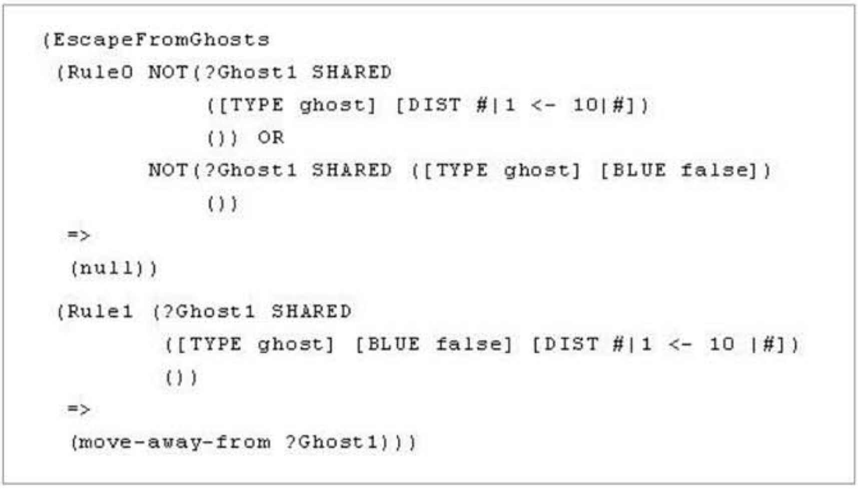
GRUE programs are "standard" teleo-reactive programs, as defined in Benson (1996), extended with resource variables. A *GRUE TRP* is a list consisting of an identifier (a string), a list of input parameters, and one or more rules. A *rule* is a list consisting of an identifier (a string), a condition, and a list of actions. The condition of a rule consists of two conjoined parts: a nonempty set of implicitly conjoined resource variables and a boolean expression of *property tests* of the form $f(x_1, ..., x_n) \Theta g(y_1, ..., y_m)$, where *f*, *g* are functions of the resource variable identifiers appearing in the condition, and Θ is a comparison operator, for example, ==, >, and <, etc. The functions include the property function as defined above, as well as user-defined functions implemented directly in the underlying implementation language. User-defined functions can use the property function to extract property values from the resource identifier. Resource variables may be negated to express the requirement that no matching resource exists. The resource variable's exclusive access flag is irrelevant in this case, and its value has no effect.

The rules are evaluated in order, with the first rule with a condition that is true proposing an action to execute. A condition evaluates to true if both the resource variables and logical expression evaluate to true. A nonnegated resource variable evaluates to true when it is successfully bound to a resource. A negated resource variable evaluates to true if no binding exists. (Binding of resource variables is discussed in more detail below.) Logical expressions containing user-defined functions evaluate in the usual way. Logical expressions may only test information derived from resource variables. For example, a rule that needs to compare the locations of two objects could contain two resource variables, one to bind each object (represented as resources), and a user-defined predicate to compare the location information extracted from each binding using the property function. If a disjunction of resource variables is required, it should be written as two separate rules.[5] Conditions are always evaluated with respect to the agent's memory, which corresponds to an agent's beliefs about the world. These beliefs are not guaranteed to be correct, for example, if the agent's sensors return partial information about the environment or the world changes between observations.

As an example, Figure 8 gives a GRUE TRP for our Pacman agent that achieves the *Escape from Ghosts* goal. When a rule condition evaluates to true, the rules actions are added to a pending actions list for possible execution. Rule actions typically change the state of the environment or the agent, and take as inputs resources or properties of resources bound in the rule condition.

In addition, the action of a rule can invoke another TRP program. This allows the agent designer to write generic programs for common subtasks that can then be used in multiple places. As an example, Figure 9 shows a simple program that moves an agent to a location or object. The calling program passes a resource that specifies the target location as an argument to the GOTO program from its resource context when it makes the call. The first rule says that if the current location of the agent matches the location property of the `?Target` resource (extracted from the resource structure by the property function), then there is nothing left to do. The second rule says that if the agent is not at the target location, then get the direction of the target, and move the agent in that direction. A call to GOTO might look like the following:

Figure 8. Example of GRUE TRP for Pacman agent



```
(EscapeFromGhosts
 (Rule0 NOT(?Ghost1 SHARED
            ([TYPE ghost] [DIST #|1 <- 10|#])
            ()) OR
        NOT(?Ghost1 SHARED ([TYPE ghost] [BLUE false])
            ())
  =>
  (null))
 (Rule1 (?Ghost1 SHARED
          ([TYPE ghost] [BLUE false] [DIST #|1 <- 10 |#])
          ())
  =>
  (move-away-from ?Ghost1)))
```

```
(Rule4 (?Enemy1 SHARED ([TYPE monster]) ()) =>(GOTO (?Enemy1)))
```

This provides the bound resource variable `?Enemy1` as the argument to GOTO. Top-level teleo-reactive programs do not take arguments.

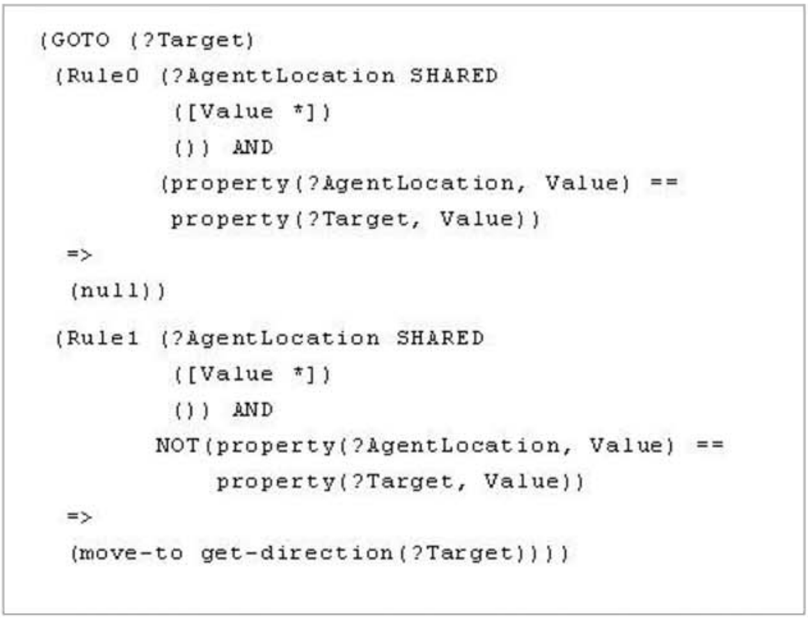
Tasks

Tasks are created from goals by the program selector. The program selector replaces the name of the GRUE TRP specified in the goal structure with the text of the TRP. The resulting data structure is a task.

A *task* is a list consisting of an identifier (a string), a priority (an integer), a type specifier (`ACHIEVEMENT` or `MAINTENANCE`), and GRUE TRP. As an example, the task in Figure 10 corresponds to the GetAwayFromGhosts goal listed above. Notice that the task identifier, priority, and type come from the goal, while the remainder of the structure is a TRP.

A task is *runnable* if the condition of a rule in the task evaluates to true. Tasks that cannot run because another (higher priority) task has preempted one or more resources are flagged as *unrunnable* and stay in the arbitrator until they become runnable. Tasks with programs that have finished executing or with programs

Figure 9. Example of program to move an agent to a location or object



```
(GOTO (?Target)
 (Rule0 (?AgenttLocation SHARED
         ([Value *])
         ()) AND
        (property(?AgentLocation, Value) ==
         property(?Target, Value))
  =>
  (null))
 (Rule1 (?AgentLocation SHARED
         ([Value *])
         ()) AND
        NOT(property(?AgentLocation, Value) ==
            property(?Target, Value))
  =>
  (move-to get-direction(?Target))))
```

that cannot run because the necessary resources are not available are removed from the arbitrator.

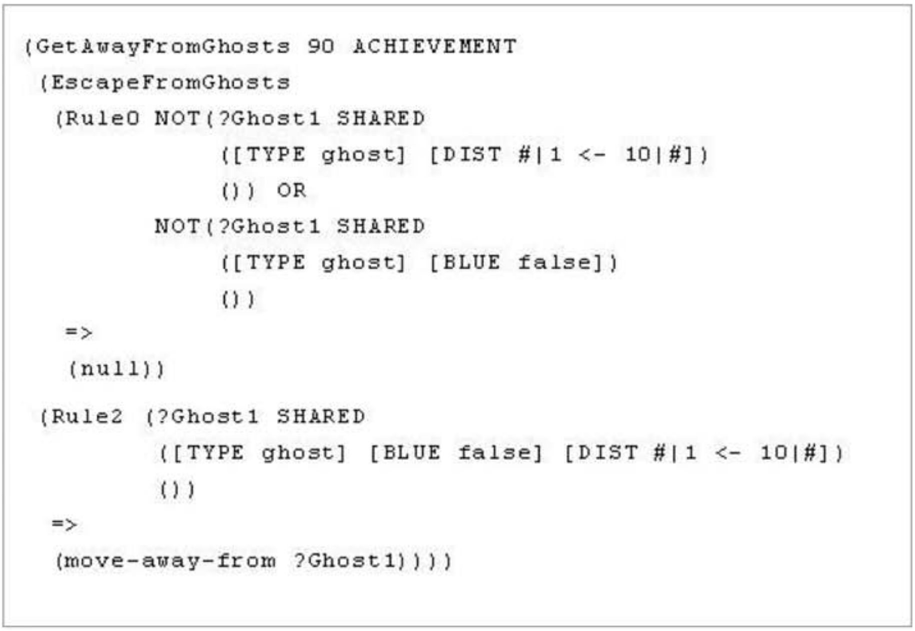
Goal Arbitration

It is the job of the arbitrator to allocate resources to the tasks, giving priority to higher priority tasks; to run the rules; and to resolve conflicts between the actions. The arbitration process allocates resources to a task by binding resource variables to available resources, then allows the task to propose an action. The list of proposed actions is examined for conflicts before the actions are executed, and conflicts are resolved in favor of the higher priority task.

Arbitration consists of five main steps:

1. Sort the tasks according to priority.
2. Starting with the highest priority task, consider the rules in textual order, looking for one that has a condition that evaluates to true.
3. If no rule in the program can run, then check whether the task should be

Figure 10. GetAwayFromGhosts specified as a task



```
(GetAwayFromGhosts 90 ACHIEVEMENT
 (EscapeFromGhosts
  (Rule0 NOT(?Ghost1 SHARED
              ([TYPE ghost] [DIST #|1 <- 10|#])
              ()) OR
         NOT(?Ghost1 SHARED
              ([TYPE ghost] [BLUE false])
              ())
   =>
   (null))
 (Rule2 (?Ghost1 SHARED
         ([TYPE ghost] [BLUE false] [DIST #|1 <- 10|#])
         ())
  =>
  (move-away-from ?Ghost1))))
```

removed from the arbitrator. This requires examining resources that have already been bound, in order to determine whether any of them would allow the task to run. If not, the task is removed.

4. Repeat Steps 2 and 3 until all tasks have been processed or the maximum number of runnable tasks is reached.
5. Execute a compatible subset of the actions proposed by the runnable rules.

A rule condition evaluates to true if there is a consistent binding of its resource variables and the associated boolean expression of predicates defined on resource property values evaluates to true. The main criterion used for binding resource variables is that a lower priority task may never take resources from a higher priority task. We therefore allow the highest priority task to bind its resources first. Unlike other rule-based languages, variable binding in GRUE is a potentially destructive operation that may change the contents of the agent's memory. If a mutually exclusive resource variable matches a resource, the resource is effectively consumed and is not available to match resource variables in other programs, though the resource may match other (EXCLUSIVE or SHARED) resource variables in the same rule. However, in cases where a resource is divisible, the EXCLUSIVE flag gives the program containing the resource variable mutually exclusive access only to the portion of the resource that it actually binds. When the property matching a value range has the DIV keyword, the resource variable will bind the optimal amount according to the utility ordering in the range. If there is no utility ordering, the minimal amount is bound. Any remainder is treated as a separate resource and returned to memory for other resource variables to bind. When a resource variable has a SHARED flag and the matching resource has the DIV keyword, the resource variable will bind the required amount, and the remainder will be returned to memory for use by other resource variables, as with an EXCLUSIVE resource variable. However, in this case, the portion bound to the SHARED resource variable also remains available for other resource variables to use.

We require that resource binding be consistent from cycle to cycle: a resource variable bound during one execution cycle remains bound to the same resource until the condition of a prior rule in the TRP matches or the resource is no longer available.

Note that no attempt is made to maximize the number of tasks that can run; in particular, the preferred properties of a higher priority task may preempt the resources of a lower priority task, preventing the lower priority task from running. Conversely, no attempt is made to limit the number of tasks the agent will run in parallel. If necessary, the arbitrator can be limited to processing only a small set of tasks. Once the task limit has been reached, the rest of the tasks are simply not run.

Tasks that are runnable in principle, i.e., could run if a higher priority task had not bound some resource(s), are not removed from the arbitrator and simply wait until they have the necessary resources to run. Tasks are removed from the arbitrator when the resources required to execute the task do not exist. In general, we would expect this to be a rare occurrence, as the goal generators and program selector should prevent programs with unfulfilled prerequisites from entering the arbitrator. The exception is the case where a required property is deleted (by the world model or another TRP) while the program is running.

The actions of the first runnable rule in each runnable task are then collected in a proposed actions list. Any task with an ACHIEVEMENT type specifier with a top rule that is runnable (i.e., propose a null action indicating that the task has been completed) is removed from the arbitrator. However, a MAINTENANCE type specifier tells the arbitrator that the task should not be removed and should continue using resources even if the goal condition is true. The remaining actions are then checked for conflicts. For example, two tasks might propose moving in opposite directions. Such conflicts are application dependent—in any given environment, some actions will conflict, while others can be executed in parallel.

The final stage of the GRUE arbitrator checks the list of proposed actions against a list of conflicting pairs of actions. If a pair of actions in the proposed actions list conflicts, the action proposed by the lower priority task is discarded. The remaining actions are then executed, changing the state of the environment or the agent, and the whole cycle starts over.

Related Work

While our work has been developed in the context of computer games, it has strong similarities to work done in other problem domains. It extends the teleo-reactive programs of Benson and Nilsson (Nilsson, 1994; Benson & Nilsson, 1995), which were developed for use in robotics. We chose TRPs as a starting point in order to take advantage of its mechanism for handling dynamic environments, as this is a key problem in both real and simulated worlds. Our work also has similarities to the cognitive architecture developed by Wright (1997), and to the BOD approach developed by Bryson (2001). In this section, we first briefly outline the differences between our work and the teleo-reactive framework proposed by Benson and Nilsson. We then discuss other similar approaches, such as those taken by Bryson and Wright.

The main difference in our use of TRPs is that we have written them in terms of resources. This gives us two advantages over the approach described in Benson and Nilsson (1995). First, disjunctive conditions can be used to represent situations where there are several alternate ways of achieving the same goal. As discussed above, there are potential problems when using stable nodes with disjunctive conditions.[6] GRUE eliminates these problems by not using stable nodes and by disallowing disjunctions involving resource variables. Instead, resource variables implicitly encode disjunctions through the use of preferences. The second advantage of GRUE is that using resources allows us to use the same basic programs for several agents but differentiate them by giving them preferences for different types of items. This property is particularly useful in entertainment applications like games.

GRUE also differs from the other approaches (Benson & Nilsson, 1995; Benson, 1996) in that the arbitrator does not use any idea of reward or time estimates. Instead, goals in GRUE are given a priority value by the goal generator. A disadvantage of using a reward/time ratio is that the programmer cannot force the agent to work exclusively on a high priority goal. If a less important goal can be completed in a sufficiently short time, it will always be completed. By contrast, GRUE's arbitrator will execute the appropriate actions simultaneously, if possible, and otherwise focus on the highest priority goal. GRUE can also be made to exhibit similar behaviour to that described in Benson and Nilsson (1995) by creating goal generators that use a reward/time ratio to compute the priority. We do not have a human user to provide rewards, but changes to the environment and the agent state could be regarded as rewards (and are represented as such by the world interface).

The architecture that is probably most similar to ours is Wright's MINDER1 (Wright, 1997). MINDER1 uses a library of teleo-reactive programs and generates and manages motives, which seem to be the functional equivalent of goals. Each motive contains a condition to be made true, an insistence value that represents the importance of the goal, and a flag indicating whether or not the motive has passed through an attention filter. The filter uses a simple threshold function, allowing motives through when their insistence is above the threshold. MINDER1 is based on a three-layer architecture developed as part of the Cognition and Affect project by Sloman and Beaudoin (Wright et al., 1996). As such, it includes both a management layer and a meta-management layer. The management layer can suspend tasks, schedule tasks, and expand a motive into a plan. The meta-management layer is responsible for monitoring the management layer and making adjustments as necessary. To make this clearer, we will explicitly compare the functions of Wright's architecture to those of GRUE. First, Wright includes an interface to the agent's sensors and a belief maintenance system that is equivalent to the world model. Next, there is a collection of "generactivators" that produce motives. Motives serve the same purpose in Wright's architecture as goals do in ours, and the generactivators are equivalent to our goal generators. Motives that pass the filter are processed by the management layer. The management layer has three tasks. The first is to decide whether or not the motives that have passed the filter should continue being processed by the management layer. The second is to determine which of the motives should be activated. Only one motive is activated at a time. Finally, it expands the motive into a complete plan by retrieving a plan from a plan library. The management layer in MINDER1 incorporates the function of our program selector and some of the functionality of the arbitrator. However, MINDER1 can only execute one plan at a time. The GRUE arbitrator has the ability to run multiple tasks at once, balanced with a mechanism for allocating resources based on dynamic priorities. Finally, MINDER1 includes a meta-management layer. This layer has two functions within the architecture. First, it can adjust the threshold in the filter. Second, it can detect perturbant states in which resources are continually diverted to particular, less useful goals. Wright's purpose in constructing MINDER1 was to investigate emotional states, some of which manifest as perturbances. We have not included a meta-management layer in

GRUE; however, it would be possible to change the filter threshold dynamically. We are not attempting to model emotional states and do not anticipate problems with perturbances.

Bryson's work (Bryson, 2001) is closely related to teleo-reactive programs. Bryson described an approach to building behaviour-based agents called behavior-oriented design (BOD). She discussed both a development process and a modular architecture that includes Basic Reactive Plans as one of the core components. Basic Reactive Plans are identical to teleo-reactive programs, with the exception that the regression property is not required. She has used BOD systems for a number of applications, including as a robot control system, to model primate behaviour, and as characters in an interactive virtual world. One significant difference between Bryson's system and ours is that her agents are controlled by drive collections. A drive collection is a Basic Reactive Plan that contains a list of tasks the agent might want to do. Because these are similar to teleo-reactive programs, tasks always have the same relative importance. For example, if an agent using a drive collection played Pacman, it might list *Escape from Ghosts*, then *Gain Points*. The goals would always be considered in that order. In contrast, a GRUE agent playing Pacman could make the importance of *Escape from Ghosts* proportional to the distance of the nearest ghost. This means that the *Gain Points* goal might be generated with a higher priority than *Escape from Ghosts*. This allows for greater flexibility in decision making.

Our architecture is not intended to be a cognitive model, but it is worth noting that some capabilities included in GRUE, particularly arbitration between multiple competing goals, are not found in most cognitive models. Both SOAR and ACT-R use only a single goal hierarchy (Johnson, 1997). Nonetheless, much work in games and real-time simulations has been done using the SOAR architecture (Laird & Duchi, 2000; Laird, 2000; van Lent et al., 1999). There have been some attempts to use multiple goal hierarchies in SOAR, however, they require either representing some goals implicitly or forcing unrelated goals into a single hierarchy (Jones et al., 1994).

DePristo and Zubek (2001) described a hybrid architecture used for an agent in a role-playing game. This type of game typically involves tasks like buying items and choosing which items should be kept in a limited inventory. The architecture included a deliberative truth maintenance and reasoning component along with a reactive layer capable of handling urgent situations without input from the deliberative layer. The system had difficulty representing quantities like amounts of gold, and because the system was focused on facts, there were some problems handling goals. In contrast to DePristo's and Zubek's architecture, GRUE is primarily focused on goals and resources rather than facts. Furthermore, we designed our data structures specifically to allow reasoning about quantities and properties of objects.

Conclusion and Future Work

In this chapter, we proposed a novel goal-processing architecture that allows an agent to arbitrate between multiple conflicting goals. Our architecture, GRUE, is based on teleo-reactive programs, which are designed to handle changes in the environment gracefully. We have shown that the teleo-reactive architecture described in Benson and Nilsson (1995) and Benson (1996) has several limitations. In environments that have a small number of stable nodes, it can become "trapped" by a single high priority goal, with the result that it is unable to respond to changes in the environment affecting the execution of other active goals. In addition, the original teleo-reactive architecture did not address the problem of goal generation and was incapable of executing multiple actions in parallel. To overcome these limitations, we introduced the idea of a resource representing a condition that must be true for the safe concurrent execution of a durative action and briefly outlined a goal arbitration scheme for teleo-reactive programs with resources that allows an agent to respond flexibly to multiple competing goals with conflicting resource requirements.

We have several ideas about possible future directions for this work. In particular, we expect it will be possible to follow Scott Benson's lead in adding planning and learning capabilities to the architecture (Benson, 1996). We also think there are some interesting possibilities in dynamic filter adjustments and the addition of a purely reactive layer.

The program selector is currently a placeholder for a full planner. To add a planning system would only require a small change. The main issue with planning is the fact that durative actions may have different effects depending on the amount of time they are allowed to operate. Benson and Nilsson solved the problem by using teleo-reactive operators (TOPs) (Benson & Nilsson, 1995), which could be treated by the planner like discrete actions. They are designed to be similar to STRIPS operators, listing the intended effect of the action as well as possible side effects.

GRUE adds the additional functionality of resource variables to TRPs. This means that preconditions for operators must include resource variables with required properties. If the same action causes different effects when used with different resources, then there must be a set of operators, each with a different list of required properties for its resource variables and the corresponding list of effects.

In Benson and Nilsson (1995), the system learned TOPs through trial and error. Again, a similar approach could be employed with GRUE. In addition, if the system has a model of emotions, then it could be set up to learn preferred properties as well as required properties. Required properties in this case are those that are necessary for an action to have the desired effect, while preferred properties are those that are most often associated with a positive emotional state. Using this method, the agent could develop preferences that, while inconsequential in and of themselves, would give the agent personality.

GRUE contains a simple threshold filter that eliminates low priority goals. This filter could be dynamically adjusted depending on the agent's current set of tasks. If the agent is working on a very important goal, it could raise the filter threshold so that only extremely important things would get through. This would, in effect, enable the agent to concentrate on something. Likewise, the threshold could be lowered to achieve the effect of a flighty agent concerned with trivialities. Dynamic adjustment could potentially give the agent moods, and use of a more complicated filter might create even more interesting effects.

Those working in biological modeling may be bothered by the hybrid approach we have taken with GRUE. While the teleo-reactive programs respond to changes in the environment, it is true that there is no way of bypassing the arbitrator. We have not as yet encountered any situations where bypassing the arbitrator would be necessary, and the overhead is small enough that we do not expect it to be a problem. However, it would be possible to add a pure reactive layer in the form of an additional ruleset that runs all the time. In order to completely eliminate overhead, it would need to bypass the main portions of GRUE by running in a separate thread or process. Doing this would potentially lead to conflicts between actions, as it would also bypass GRUE's conflict resolution mechanism. This problem could be handled by requiring that the reactive layer contain only actions that do not conflict with any other actions, or the potential conflicts (and resulting confusion) might be considered acceptable by the agent designer.

Acknowledgments

Elizabeth Gordon is a PhD student supported by Sony Computer Entertainment Europe. Thanks also to Jamie Strachan for assistance in proofreading the chapter.

References

Benson, S. (1996). *Learning action models for reactive autonomous agents*. PhD thesis, Stanford University.

Benson, S., & Nilsson, N. (1995). Reacting, planning and learning in an autonomous agent. In K. Furukawa, D. Mitchie, & S. Muggleton (Eds.), *Machine intelligence* (Vol. 14) (pp.29-64). Oxford: Clarendon Press.

Bryson, J. J. (2001). *Intelligence by design: Principles of modularity and coordination for engineering complex adaptive agents*. PhD thesis, MIT, Department of EECS, Cambridge, MA. [AI Technical Report 2001-003.]

DePristo, M., & Zubek, R. (2001). Being-in-the-world. In *Proceedings of the 2001 AAAI Spring Symposium on Artificial Intelligence and Interactive Entertainment* (pp.31-34). [Technical Report SS-01-02]. AAAI Press.

Freed, M., Bear, T., Goldman, H., Hyatt, G., Reber, P., & Tauber, J. (2000). Towards more human-like computer opponents. In *AAAI 2000 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment* (pp.22-26), March 2000. [Technical Report SS-00-02]. AAAI Press.

Hawes, N. (2000). Real-time goal orientated behaviour for computer game agents. In *Proceedings of Game-ON 2000, First International Conference on Intelligent Games and Simulation* (pp. 71-75). Society for Computer Simulation International.

Johnson, T. R. (1997). Control in act-r and soar. In M. Shafto & P. Langley (Eds.), *Proceedings of the 19th Annual Conference of the Cognitive Science Society* (pp. 343-348). Mahweh, NJ: Lawrence Erlbaum Associates.

Jones, R., Laird, J., Tambe, M., & Rosenbloom, P. (1994). Generating behavior in response to interacting goals. In *Proceedings of the Fourth Conference on Computer Generated Forces and Behavioral Representation* (pp.317-324), Orlando, Florida.

Kaebling, L. P., & Rosenschein, S. J. (1994). Action and planning in embedded agents. In P. Maes (Ed.), *Designing autonomous agents* (pp.35-48). Cambridge, MA: MIT Press.

Laird, J. (2000). It knows what you're going to do: Adding anticipation to a quakebot. In *AAAI 2000 Spring Symposium Series: Artificial Intelligence and Interactive Entertainment* (pp.41-50), March 2000. [Technical Report SS-00-02]. AAAI Press.

Laird, J., & Duchi, J. (2000). Creating human-like synthetic characters with multiple skill levels: A case study using the soar quakebot. In *AAAI Fall Symposium Series: Simulating Human Agents* (pp.75-79). [Technical Report FS-00-03]. AAAI Press.

Laird, J. E., Congdon, C. B., Altman, E., & Doorenbos, R. (1993). *Soar User's Manual, Version 6* (1st ed.).

Nilsson, N. (1994). Teleo-reactive programs for agent control. *Journal of Artificial Intelligence Research, 1*, 139-158.

van Lent, M., Laird, J., Buckman, J., Hartford, J., Houchard, S., Steinkraus, K., & Tedrake, R. (1999). Intelligent agents in computer games. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence* (pp.929-930). AAAI Press/MIT Press.

Wright, I. (1997). *Emotional agents*. PhD thesis, University of Birmingham.

Wright, I., Sloman, A., & Beaudoin, L. (1996). Towards a design-based analysis of emotional episodes. *Philosophy Psychiatry and Psychology, 3*(2), 101-137.

Endnotes

[1] Nilsson noted that several other systems based on control theory and electrical circuitry have been proposed, primarily Kaebling's GAPPS system (Kaebling & Rosenschein, 1994), but asserts that these systems, because they are precompiled, may construct extra circuitry. In contrast, TRPs construct circuitry at runtime, thus creating only what is needed.

[2] Small increases in the time taken to respond to events may actually result in increased believability (Freed et al., 2000).

[3] The resulting resource structure is similar to the augmentations of SOAR's working memory elements (Laird et al., 1993), except that here, values cannot be other resources, only constants.

[4] This is assuming that the goal generators never fire when the condition of the goal they generate is already true.

[5] Some disjunctions can be encapsulated in a single resource variable, through the judicious use of preferred properties.

[6] Disjunctive conditions are not discussed in Nilsson (1994), Benson and Nilsson (1995), or Benson (1996). However, both Benson and Nilsson (1995) and Benson (1996) contain examples of disjunctive conditions. It is possible that the problems mentioned here were simply not encountered in the environments used in Benson and Nilsson (1995) and Benson (1996).

Chapter 12

Artificial Minds and Conscious Machines

Pentti O. Haikonen
Nokia Research Center, Finland

Abstract

The following fundamental issues of artificial minds and conscious machines are considered here: representation and symbolic processing of information with meaning and significance in the human sense; the perception process; a neural cognitive architecture; system reactions and emotions; consciousness in the machine; and artificial minds as a content-level phenomenon. Solutions are proposed for related problems, and a cognitive machine is outlined. An artificial mind within this machine that eventually controls the machine is seen to arise via learning and experience as higher level content is constructed.

Introduction

Could a robot have a conscious mind? We humans are superior to machines and computers because we can think, we can understand, and we are conscious. We humans perform actions that are meaningful to us, ourselves. The actions of present-day machines may be meaningful to us but not to the machine, because we have a mind, while the machine has not. Therefore, to create higher machine cognition, we should investigate the possibilities of true thinking and conscious machines, ones with artificial minds. These machine minds should imitate the human mind in all its expressions and ways of operation in order to achieve human-level mental performance.

The human mind is characterized by the flow of inner imagery, inner speech, sensations, emotional moods, and the awareness of these. The human mind is imaginative, creative, and intelligent. The human mind possesses intentionality; it operates with meanings and significance, and it understands what it is doing. The human mind seems to effortlessly unify past experience; present multisensory information; the expected and desired future; needs, drives, and goals; as well as any triggered emotional states. Our minds have contents that make us what we are—our personal history, fortunes and misfortunes, needs, values, secret desires, yearnings and hopes, and our ways of perceiving and acting upon the world. Mind and consciousness go together. Consciousness provides us the instantaneous focus of awareness and the ability to report our mental content to ourselves and others. The creation of conscious machines is the ultimate challenge of artificial intelligence (AI).

Background

The last decade of the 20th century saw a growing interest in machine cognition and artificial minds. At the beginning of the decade, Trehub (1991) proposed a neural network scheme for cognition but did not address the question of consciousness. Taylor (1992, 1997, 1999) and Aleksander (1993, 1996) began to propose neural mechanisms for consciousness and mind. Duch (1994) tried to find a neural/symbolic approach to artificial minds. Valiant (1994) studied some possible algorithms for the mind. Sloman (2000) studied the requirements for minds on a more general level. More philosophical approaches were presented by, for example, Dennett (1991) and Baars (1997). Also, in the same decade, the author began his work on machine cognition (Haikonen, 1994, 1998a, 1998b, 1998c, 1999, 2000a, 2000b). Recent advances in computing power allow the simulation of complex neural networks. An example of this approach is the animated *CyberChild* simulation of Cotterill (2003). A traditional software program approach is also possible, like the "conscious" software agent IDA of Franklin (2000, 2003). Artificial minds and robotics go naturally together. Holland (2003), for example, has tried to provide robots with some kind of consciousness via internal models.

Artificial conscious machine minds should model human cognition and consciousness. This might be done by software. It can also be argued that only dedicated hardware systems with system reactions can truly capture the essence of consciousness. This approach has been proposed by the author (Haikonen, 2003) and is elaborated here.

Problems to be Solved

The fundamental issues of artificial minds and conscious machines are as follows:

1. The representation problem. How can information be represented by neural symbols with meaning and significance in the human sense?
2. The perception problem. The perception process provides the machine with information about external and internal conditions. A simple model of perception as pattern recognition and classification is hardly satisfactory. What would be a better solution?
3. The architecture problem. What kind of architecture could combine sensory information with the system's present status, past experience, needs, and goals and support the flow of inner imagery, inner speech, system reactions, and emotions?
4. Emotions and values. Are these needed? How could they be implemented?
5. The problem of consciousness. Can a machine be conscious?
6. The problem of mind. It is proposed here that a mind, human or artificial, is a content-level phenomenon. How could a mind-like content accumulate in a machine?

In the following, these issues are discussed, and some solutions are proposed.

Representation, Meaning, and Signal Arrays

A cognitive system derives its information about the external world from sensors. There are numerous ways to represent sensory information. The information from a microphone takes the form of time-varying voltage. The information from a state-of-the-art image sensor takes the form of picture element intensity values, which again may be represented by voltage values. But, what external world entities would these voltage signals represent? The problem here is not insufficient information but rather the opposite. The variations of these signals may be practically infinite, and therefore, consistent associations with external world entities would be next to impossible. Speech recognition demonstrates this problem. A speaker may have a higher or lower speaking voice, he or she may speak faster or slower, he or she may speak aloud or whisper, there may be background noise, yet the words should be recognized as the same. Should the recognition system learn every possible signal appearance for each word? This is not really practical, and obviously, some invariant features should be looked for. This calls for signal preprocessing, spectral analysis, the detection of the relative intensities of harmonics, etc. Likewise, visual information, in the form of picture element value matrices, is not useful as such. A visual object will not produce every time exactly the same picture element excitations on the image sensor, as the viewing distance, angle, and illumination may change. Therefore, invariant visual features that survive viewing angle and illumination changes should be looked for.

In a computer environment, sensory information is temporally sampled, digitized, and represented as series of binary numbers. However, the brain is not a numeric calculator of binary sequences, and cognitive machine designs that seek to imitate the brain should utilize other representations. The principles of distributed representations (Hinton, McClelland, & Rumelhart, 1990) are relevant here. Therefore, nonnumeric distributed signal representation has been chosen for the author's design approach.

Distributed signal representations consist of large signal arrays. In this context, the initial meaning of each signal is derived from the corresponding sensor. Each of these signals represents a preprocessed fraction or feature of the total information available from that sensor. Thus, a given signal array will represent a combination of invariant features.

Distributed signal representations can also be used to control motor acts. A motor act sequence can be realized as a serial and parallel combination of elementary movements, motor primitives, each controlled by a single signal. Each composite motor act can be governed by a controlling sequence of signal arrays, a sequence of distributed representations that switches motor primitives on and off as needed.

In the machine, a visually sensed object will cause an array of signals that corresponds to the collection of the abstracted features of the visual pattern; a sensed sound will cause a sequence of signal arrays that corresponds to the collection of the components of the auditory pattern; and an intended motor act will initiate a sequence of signal arrays that corresponds to the motor primitives in the intended sequence.

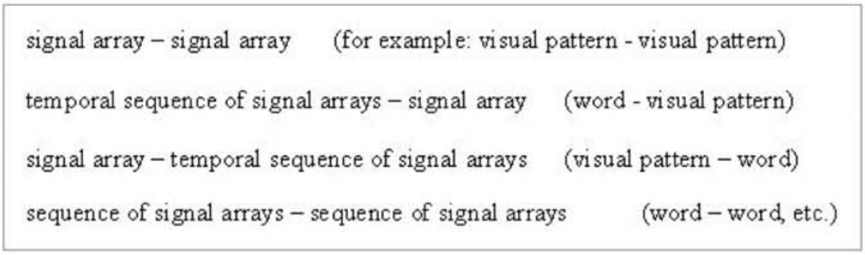
However, while all this is trivial and sufficient for simple stimulus–response systems, it will not allow higher cognition and thinking. Thinking involves the use of symbols that depict entities, actions, relationships, etc. In the proposed representation method, only signal arrays exist, and therefore, only these could be used as symbols. A symbol must be able to evoke representations of the entities that it depicts. Likewise, an entity must be able to evoke the corresponding symbol. This can be achieved via association. For the creation of such associations, the intended symbol must first appear in the form of a signal array. Signal arrays are not expected to pop out of nothing, and it is not implied that a human operator should inject these into the system. The only way that a repeating signal array can appear in this artificial system is via sensory stimulation. Therefore, the symbols would have to have the appearance of a sensed entity. This seems to be the case with human cognition, too. Words, for instance, are actually sound patterns. Likewise, letters and numbers are visual patterns. However, it should be noted that the perceived appearance of a symbol may not have anything to do with the symbolized entities. In this context, a symbol is not necessarily an *analog image*, and the proposed representation method does not rely on any analog properties of the symbols.

It is important to note that in the proposed representation system, different categories of representation exist: ones that are component representations of sensed entities and ones that besides being that are also higher level symbols. Both cases appear physically as signal arrays and sequences of signal arrays. Barsalou (1999) has argued that cognition must be based on perceptual symbols, which are modal and analogical, componential and not discrete. Here the component representations of the sensed entities would seem to fit this description. On the other hand, AI utilizes symbols that are apparently nonperceptual. Here, the meaning of higher level symbols is no longer limited to their sensory signal origin. Therefore, these symbols can be used to represent abstract entities and their relationships.

Association between representations that are carried by signal arrays and sequences of signal arrays will be a necessary mode of processing. This association will allow the evocation of a given representation by the associated representation. The associative linking capacity shown in Figure 1 would be especially needed.

Distributed representations, arrays of signals, are not monolith symbols. Instead, they have an inherent fine structure that allows response evocation by partial representations, too. This leads to automatic classification—representations that are similar enough can evoke the same response representation. No extensive learning is needed, as only one example with an associated response will do, and

Figure 1. Required associative linking capacity



signal array – signal array	(for example: visual pattern - visual pattern)
temporal sequence of signal arrays – signal array	(word - visual pattern)
signal array – temporal sequence of signal arrays	(visual pattern – word)
sequence of signal arrays – sequence of signal arrays	(word – word, etc.)

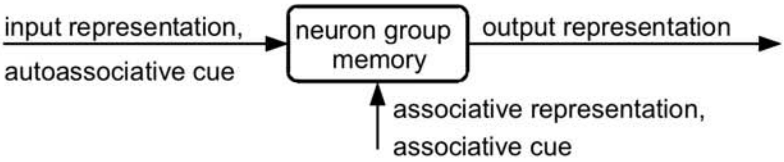
all further representations that are close enough to the original will evoke the same response.

The author has designed and experimented with artificial associative neuron group microchips that are able to learn the required associations via a modified Hebbian process (Rantala & Haikonen, 2002). Here, the associative connection between two signals is established if these signals appear together repeatedly. Time-averaging is also included; if the signals fail to coincide, the eventual associative connection is impeded. Thus, random coincidences will not accumulate or become learned associations. This mechanism also allows for the labeling of a property that does not appear separately. A large number of these neurons can be used to learn associations between signal arrays. Additional short-term memory elements are needed for the manipulation of sequences.

It suffices here that an artificial neuron group is defined that is able to learn and produce the above-mentioned associations between signal arrays and sequences of signal arrays.

The neuron group of Figure 2 has two inputs, namely, the input representation and the associative representation. Both inputs are signal arrays. There is one output: the output representation. The categorical meaning of the output representation is the same as that of the input representation. The neuron group is able to execute auto-association in which the input representation is associated with itself. Thereafter, a part of an input representation can be used as an auto-associative cue that will evoke the complete representation, which will then

Figure 2. Defined neuron group



emerge as the output. The neuron group is also able to execute cross-association, in which the associative representation is associated with the input representation. Thereafter, the associative representation can be used as a cue that evokes the previously associated input representation or sequence as the output. In this way, the generic neuron group works as an associative (episodic) memory that responds to partial evoking representations. In practice, the associative evocation process may try to evoke a large number of signals at various signal levels. Of these, only the strongest are relevant and can be selected by means of an automatically set threshold. In this way, the winner-takes-all principle can be applied. This principle can also be utilized at the associative inputs. If competing evoking representations occur, the strongest evoking representation will win, and its response will emerge as the sole output. However, the recalled sequence of output representations is a reconstruction, and as such, may or may not be an exact copy of any original memorization.

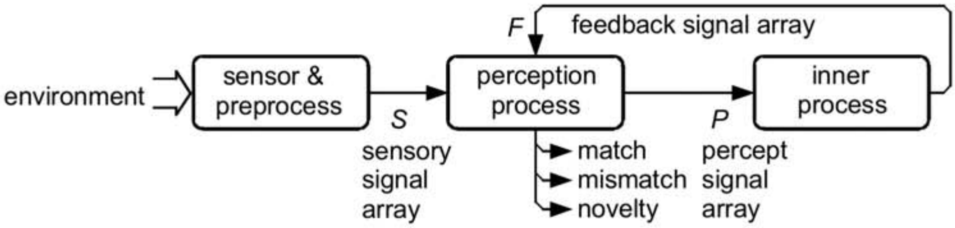
Machine Perception Processes

In this context, machine perception is defined as the process that allows the machine to access and interpret sensory information and introspect its own mental content. The interpretation of sensory information is not a matter of simple recognition and labeling. It is a wider process involving the context, experience, and needs of the system. The perceived entities should not be strictly categorized objects. Instead, they should be seen by the machine as possibilities for action afforded by the environment. A stone may be taken as a hammer if called for by the situation. A similar view of human perception as the recognition of affordances has been proposed by Gibson (1966).

Thus, the perceptual process must combine the effects of sensory information and the system's cognitive state. The system must determine which parts of the sensory information should be attended to and amplified so that relevant associations could be made. This requirement calls for feedback from the system's inner processes. An outline of a simple perception system with feedback, the perception loop, is depicted in Figure 3.

The perception loop performs the functions of sensory perception, introspection, and reverberating short-term working memory. In Figure 3, the sensory information is preprocessed, and a flow of distributed signal arrays is generated. These signal arrays must represent generalized features of the sensed entities. The perception process combines the effect of sensory signal array *S* and the internally generated feedback signal array *F*. The resulting signal array *P* is called a percept signal array, as it will now be the official output from the

Figure 3. Perception loop

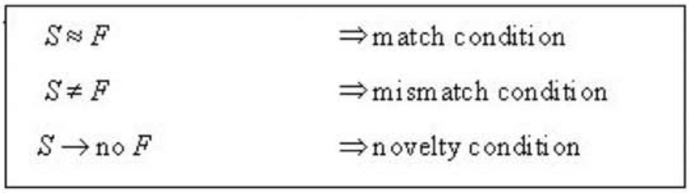


perception process and, as such, will be forwarded to the system's inner processes. The perception process also determines match, mismatch, and novelty conditions between the sensory array *S* and the feedback array *F*. The sensory match, mismatch, and novelty conditions are defined as shown in Figure 4.

Here, the *S* and *F* arrays must represent the same category. For instance, if *S* represents color, then *F* must represent color, too. It should be noted that in any case, *S* and *F* must represent extracted generalized features that allow feasible comparisons. For general cognitive purposes, it would not be very useful to try this comparison with, say, raw picture element data, as the match-condition would rarely occur.

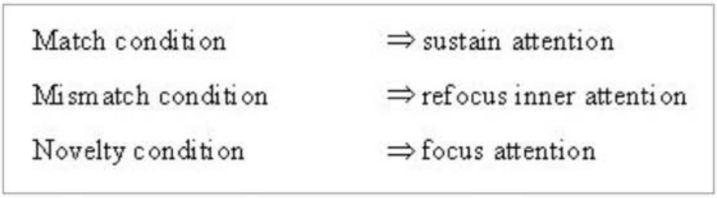
The feedback signal array may constitute a prediction for the sensory signal array. In that case, the match/mismatch/novelty condition would indicate the success of the prediction. Sometimes the feedback signal array may depict an entity to be sought. In that case, the match/mismatch condition would indicate the success of the search. It is obvious that the system should strive toward the match-condition. This it can do through attention control. The desired effect of match/mismatch/novelty conditions on attention can be summarized as shown in Figure 5.

Figure 4. Sensory matching, mismatching and novelty conditions



$S \approx F$	$\Rightarrow$ match condition
$S \neq F$	$\Rightarrow$ mismatch condition
$S \rightarrow$ no F	$\Rightarrow$ novelty condition

Figure 5. Effect of matching conditions on attention



Match condition	⇒ sustain attention
Mismatch condition	⇒ refocus inner attention
Novelty condition	⇒ focus attention

The feedback mechanism also facilitates introspection. Introspective perception of inner imagery or other inner representations takes place when the percept is caused by feedback signals only.

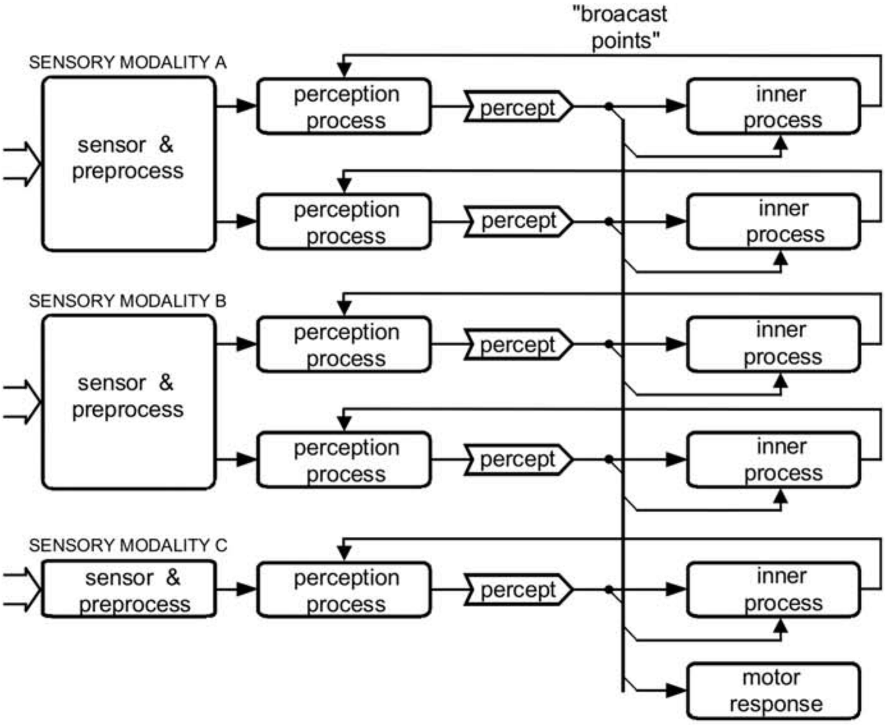
Cognitive Architectures: Platforms for Artificial Minds

A prerequisite for an artificial mind is the combination of sensory information from multiple sensory sources and the machine's own knowledge, including the machine's present status, needs, and goals. The following functions need to be integrated: multisensory perception, attention, short-term and long-term memories, learning, inner speech and imagery, judgement, deduction, reasoning, planning, motivation, response generation, system reactions, and emotions. The system must also combine visual, auditory, haptic, and egocentric (body position) information so that a consistent view of the environment can be achieved—one that allows for instance fluent navigation and object interaction.

Scene and episodic understanding involves the determination of what is where, what has changed, what is the action, who is doing what, etc. The perceived entities must evoke a variety of associations, like relationships, possibilities for use and action, predictions of futures, good/bad and importance evaluations, etc. Obviously, these operations call for structured information processing that goes beyond the basic functions of the elementary circuits introduced so far. Therefore, they can best be solved by means of a suitable system architecture.

The reconstruction of a cognitive architecture may begin with the previously described perception loops. In a system with multiple sensory modalities, each modality would have its own perception loops. The system would also include a motor response module. It is obvious that each sensory modality would need access to the motor response module. Motor responses could be needed for

Figure 6. A cognitive architecture based on cross-coupled perception loops



visual, auditory, tactile, etc., stimuli independent of each other. Associative connections between the perception loops would also be required. This leads to a massively parallel modular system with global broadcasting and cross-communication of instantaneously attended content. The author has proposed architectures that potentially satisfy these requirements (Haikonen, 1999, 2003). An outline of this kind of system architecture is depicted in Figure 6.

The cognitive system of Figure 6 consists of a number of sensory modalities. Each sensory modality has a very large number of the perception loops typified in Figure 3. The total system may have thousands or even millions of these loops; only a few are depicted here for the sake of clarity. Each loop produces its own percept, which is the instantaneous official output of that loop. This percept is then broadcast to the inner processes of other loops. The broadcasting takes place via associative coupling. This coupling will allow the association of percepts from different modalities with each other, so that later on, one can be evoked by the other. Thus, for instance, the sound of a telephone may evoke visual imagery of the same and may further evoke motor sequences that allow the reaching out for it. The evoking percept does not have to be generated by sensory stimuli. Instead, it may consist of feedback from the inner processes. Thus, sequences of *imagined* percepts may be generated, and possible motor responses for these may or may not be acted out.

Everything should not be connected to everything, though. For instance, color signals should be connected only to a small group of neurons at the auditory inner process. These neurons would then come to represent color-related words. Likewise, other category words would be represented by other distinct neuron groups. This approach minimizes cross-connection wiring and interference from extensive overlapping of signal arrays.

Every broadcast percept cannot be a winner. Inner process blocks have variable input thresholds that filter out weak broadcasts, and also stronger ones if the receiving module is busy with its own content. For example, a motor response module may not need input from other modules during the execution of a routine act (like walking, etc.). In fact, some input might deteriorate the execution of the act, as the timing, etc., might be disturbed by the input. In other cases, the broadcast percepts would be about the same entity and would represent its different properties as well as related motor responses. In this case, the system's inner attention would be globally focused on that entity.

The author has experimented with a subset of this architecture, which was realized as a computer program with a real-world video camera. It was shown that this kind of architecture can learn simple natural language with meaning grounded to external entities as well as to other words (Haikonen, 1999). Visually perceived entities and actions can be labeled. New visual entities can be learned by verbal description only. Named objects can be searched visually. The structure of a question can be learned by question-answer example sentence pairs. Thereafter, similar-style questions can be answered by looking for the answer from the visually sensed external world or from evoked inner imagery. The system is able to learn the meanings of "yes" and "no" by grounding these to match/mismatch conditions. There is *mental* content in the way of inner imagery and inner speech. The experiment did not incorporate actual motor responses or routines apart from the control of gaze direction, which was implemented in an electronic way.

System Reactions and Emotional Significance in the Machine

Traditionally, reason and emotions have been seen as opposite—emotions do not and must not have any part in logical reasoning. This may be so, yet human cognition and emotions are closely connected. The value of emotions has been pointed out by LeDoux (1996) and others; emotional significance is seen as an important criterion that guides learning and decision making (Davis, 2000; Haikonen, 2002). There are various theories about emotions, what they are, and how they operate. The two-factor theory proposes that emotions involve the perception of a triggering event, physiological reaction, cognitive evaluation, and subjective feeling (Schachter & Singer, 1962). Plutchik (1980) proposed that there are only eight basic emotions, and others are combinations of these. These and other theories of emotion offer only vague guidance to a design engineer. Therefore, the author has tried to condense the essence of these theories into a practical approach to machine emotions (Haikonen, 2003). This approach is not claimed to be psychologically accurate, but at least it is artificially implementable and leads to practical system behavior. This approach is outlined in the following.

A true cognitive system must be able to evaluate the significance of an event, giving rise to markers like "this is good," "this is bad," "this is dangerous," etc., and base its further operation on the results of these evaluations. This evaluated significance should guide attention, learning, and memorization. Biological cognition bases its significance evaluation on elementary sensations like taste, smell, pain, and pleasure. These elementary sensations are also often generalized to apply to more abstract matters. Elementary sensations may also evoke basic system reactions as shortcut responses to the situation.

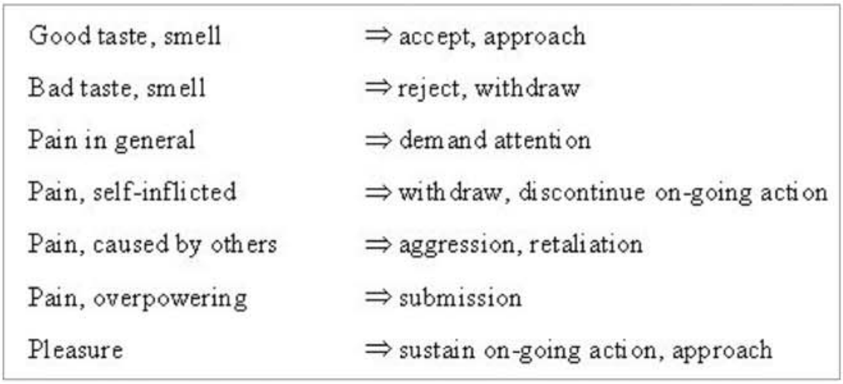
The author proposes that machine evaluation of significance should be based on elementary sensory information originating from suitable sensors. These sensors should include good and bad value input points (the equivalents for smell and taste). Likewise, the initiation of system reactions that relate to pain and pleasure may be based on artificial inputs. In robotic applications, sensors of physical damage could be used as pain sensors. These inputs could then also be used to punish and reward the system.

These are called elementary sensors, as their outputs need little in the way of cognitive interpretation. Instead, in biological systems, sensations from these sensors have direct and automatic system reactions. Some of these are listed in Figure 7.

It can be seen that these sensations have built-in judgement criteria; they dictate whether something is good or bad, and whether some activity should be continued or not. They also offer shortcut reactions as immediate responses. In human cognition, these criteria and reactions are also generalized to apply to other sensory percepts and situations. Thus, things and situations in general may acquire emotional significance. They will be pleasant and desirable or bad and to be avoided; they will evoke anger; etc.

The implementation of emotion-like states is considered in the context of associative signal array processing. The creation of machine emotions begins

Figure 7. Example direct and automatic system reactions

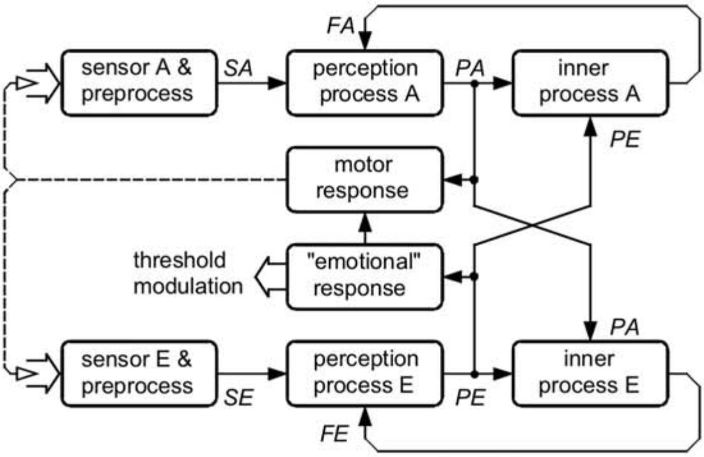


Good taste, smell	⇒ accept, approach
Bad taste, smell	⇒ reject, withdraw
Pain in general	⇒ demand attention
Pain, self-inflicted	⇒ withdraw, discontinue on-going action
Pain, caused by others	⇒ aggression, retaliation
Pain, overpowering	⇒ submission
Pleasure	⇒ sustain on-going action, approach

with emotional significance. This should be used to focus attention on the emotionally significant signal arrays and to evoke the related system reactions. These functions can be added to the basic system model in the way depicted in Figure 8.

Figure 8 depicts the outline of the principle of emotional control in an artificial cognitive system. For the sake of clarity, only two perception loops are depicted here, but an actual system would have a large number of them. Sensor **A** represents vision, audition, etc., sensors, and *PA* represents percepts from these. Sensor **E** and the related preprocess depict elementary sensors for good, bad, pain, and pleasure. *SE* designates the sensory signal array from these. The corresponding percept *PE* may initiate a respective "emotional" response that

Figure 8. "Emotional" control of a cognitive system



will control attention by modulating signal intensities and threshold levels for inner processes. The "emotional" response also affects the execution of motor responses; for example, the direction, force, and speed.

According to this model, percepts from other sensory modalities may become associated with "emotional" significance if these percepts appear together with percepts of pain, pleasure, goodness, or badness (*PA* at inner process E). Thereafter, this percept, sensed or imagined, may trigger the associated "emotional" significance and the related response. This process may even be abstract, as the emotional percepts (*PE*) may be evoked by inner causes only or by verbal description. Thus, for instance, the system may be verbally taught that "this is bad," etc.

Typical human emotions, like fear, anger, hate, envy, desire, etc., each suggest certain categories of responses—should something be approached or avoided, should the reaction be aggressive or submissive, etc. In this way, emotions function as kinds of model templates for action. This template may not be unequivocal, as emotions often involve conflict and confusion. Desire may be sustained because the object of the desire cannot be approached. Astonishment involves indecision between approach and withdrawal, etc. Emotions function as important motivational factors. We do things out of curiosity, fear, anger, envy, jealousy, guilt, expectation of pleasure or punishment, etc. Human emotions also have a subjective aspect—they feel like something and have physiological effects.

The author proposes that the functional aspects of emotions could be implemented in a machine as combinations of system reactions caused by actual and generalized elementary sensations. In this way, we could, for instance, generate the system reactions of Figure 9.

Figure 9. Example emotion-based system reactions

curiosity	=	novelty + approach
astonishment	=	mismatch + approach + withdraw
caution	=	novelty + withdraw
desire	=	pleasure + good + approach
anger	=	bad + aggression
sadness	=	bad + submission

Emotional significance can be used as a motivational factor in cognitive machines, too. For instance, the emotional significance of pleasure may be associated with the execution of a given task. Thereafter, the system would focus its attention to the execution of that task at the slightest cue, whenever the environment would allow it. Likewise, the system could be made to avoid actions where the emotional significance is of displeasure or pain.

Machine emotions would involve instant judgement by emotional value, system reactions, and direction of action, as well as motivational effects. In principle, these would be useful functions but sometimes might counteract more appropriate rational responses. Quick emotional shortcut reactions in a dangerous situation may save the day for the robot, but a robot in emotional rage would be no good for most purposes. It is up to the designer to find a proper balance here.

Consciousness in the Machine

Aspects of consciousness, such as the awareness of environment, the system self, and the ability to report on these, are considered here. These aspects are seen to arise from the broadcasting of percepts across the various modalities. A percept that is to become part of the instantaneous contents of consciousness is broadcast to and accepted by most modalities and may thus be reported in terms of these modalities. Cross-associations are formed and memory traces are created at various locations. Therefore, a "conscious" percept and event may be recalled and reported afterwards. In this kind of "consciousness," the focus of each module is allocated to the "conscious" event. "Self-consciousness" would arise in a similar way from the percepts of the physical self and mental content.

How can we know if a machine is "conscious"? Perhaps some key functions could be looked for. For instance, Aleksander (2003) listed five supposedly typical properties of a conscious system: sense of place, the faculty of imagination, the ability to direct attention, the ability to plan, and emotions in the decision process.

The mental content of a supposedly conscious machine would consist of percepts about the environment, the physical status of the machine, and finally, introspective percepts of the mental content. Machine consciousness could be tested by investigating whether the machine thinks and is aware of mental content.

Human thinking is characterized by inner speech and inner imagery; a thinking machine should have these too. As the designers of the machinery, we will know whether the machine has these or not. We will also be able to monitor these in an actual running system. In the proposed architecture, inner speech is carried by the signal array flow in the auditory modality. The individual signal arrays in this flow correspond to the sensory representations of phonemes. We cannot connect these signals directly to an audio amplifier to make them audible. However, audible flow of inner speech can be synthesized from this information by the vocoder principle (Dudley, 1939). Likewise, the flow of inner imagery can be made visible even if in a sketchy way. In this way, the instantaneous active mental content of the machine can be monitored, which is a feat that is not yet available for human subjects. Thus, it would be reasonably easy to see and test whether the machine thinks in a human way by determining if it has the flow of inner speech and imagery with correctly grounded symbol utilization and attention. We could also monitor system reactions by conventional means and see if emotion-related reactions are present. However, in this way, we could not tell if the subjective feel of pain, for instance, was present.

If the machine is able to report on the perception of environment and body as well as its own inner speech and imagery and claim the ownership of these, then one aspect of consciousness is there. If, on the other hand, the machine were not able to report these, not even in principle, then the machine would not really be conscious. However, a machine could produce these reports in ways that are definitely not due to consciousness. Therefore, we would have to see that the reports were generated in a plausible way.

Artificial Minds

Suppose that a thinking machine along the ideas of this chapter has been built and is ready to be switched on. Does it have a mind now? The author does not think so. At the moment it is switched on for the first time, it will possess the faculty of perception, a collection of cognitive functions, and the "conscious style" of operation, but, unless vast a priori information is provided, it will not have much of a mind. As the author sees it, a mind, human or artificial, is not a neural network, not the hardware architecture, or a collection of cognitive functions; instead, it is a content-level phenomenon. A mind will only arise when the system learns about itself and its environment, learns to seek to satisfy its needs and drives, acquires its own beliefs and values, and begins to adjust its behavior according to these. True, there can be no mind without a supporting hardware, but the hardware and the cognitive functions provided by it will only be half of the story. Problems with the hardware may cause problems with the mind, but mind-related problems might arise in good hardware.

Thus, the creation of an artificial mind calls for the creation of supporting machinery and also the training of the system. Some parts of this training may be provided by self-learning, and some parts may require a teacher. Good and bad values will be necessary, and reward and punishment may have to be utilized. How exactly will this be done, and how much effort will be needed? Should we leave this to engineers, or should we hire an elementary school teacher? This remains to be seen, but once generated, a successfully trained system may be copied and reproduced by conventional electronics manufacturing processes.

Future Trends

The eventual construction of a conscious machine with an artificial mind will have profound philosophical significance. However, in today's world, practical applications will be important. Obvious applications will be found where human-like understanding, flexibility, and creativity are required. Artificial cognition including abilities like the machine understanding of situation, scenery, speech, and text, as well as imagination, reasoning, planning, and learning will have an important role in future information technology and robotic applications. It may well be that human-like robots, nurses, etc., that interact and work with humans will not be completely successful without artificial minds. It may also be that these kinds of robots would have to understand how the human mind works, at least to the same extent as we humans naturally do.

The exact applications remain to be seen, as for the most part, the published research on machine consciousness seems to be theoretical. Franklin's IDA, while not yet truly conscious, has a practical application, and as such, is one exception to the present situation (Franklin, 2000, 2003). It can be expected that in the near future, more of the research on conscious machines will focus on practical applications and their demands.

Conclusion

An approach toward machines with artificial minds has been outlined here. Artificial machine mind is seen here as the higher level content of the cognitive system. The machine mind would have its own knowledge, needs, goals, and values, which it would acquire mainly via experience and learning.

The author's system can be compared to other models of mind and consciousness. Aleksander (1993, 1996) has investigated machine cognition with his neural system simulation called Magnus. Aleksander utilizes a neuron that has a number of inputs. One of these inputs is a "dominant input" and, in effect, the signal patterns at the other inputs will become associated with the signal at that input. This function is, in a broad sense, similar to the neuron architecture introduced in this chapter. Aleksander's neural system consists of cross-connected state machines, assembled from the neurons. Thus, Aleksander is able to describe the operation of the system by state diagrams. It is known that a state machine does not possess full Turing power, as it cannot execute computations that require arbitrary recall of intermediate results. This can be remedied by including random access memory in one way or another. However, it is not clear if and how this has been implemented in the Magnus simulation. The author's system is not a state machine. The architecture is different, and the function of random access memory is implemented in a distributed and associative way. Aleksander emphasized the importance of machine imagination, in his case, in the form of *iconic* states. The author agrees on the importance of imagination. In the author's system, imagination is effected by the cross-coupled inner processes, and the resulting imagery signal arrays appear at the percept points.

Baars' global workspace theory proposes a "theater stage" as the site for the conscious inner imagery and inner speech (Baars, 1997). In the author's model, the percept locations may be compared to the Baars' theater stage, as they contain the inner speech and inner imagery, and these representations are broadcast to the other parts of the system. Baars proposed that this theater stage be located at the sensory projection areas, which is also the case in the author's model. However, Baars does not really explain how information should be represented by neural firings or how actual neurons should be connected into networks that would constitute a complete cognitive system.

A machine that is supposed to parallel and surpass humans in cognitive tasks must utilize methods that surpass those used by AI today. We must proceed from speech recognition to speech understanding, from pattern recognition to scene understanding, from sentence parsing to story understanding, from statistical learning to cognitive learning, and from numerical simulation to free imagination. The ever-increasing speed and capacity of the microprocessor alone will not take us there, but advances in integrated circuit technology may eventually enable us to build machines that successfully emulate human cognition — machines that have artificial minds.

Acknowledgments

The related work was done at the Nokia Research Center (NRC). The author wishes to thank the head of NRC, Dr. Tero Ojanperä, for support and the continuing possibility to work in this interesting area. Additional financial support has been provided by the National Technology Agency of Finland (TEKES), which is gratefully acknowledged.

References

Aleksander, I. (1996). *Impossible minds my neurons my consciousness*. London: Imperial College Press.

Aleksander, I., & Dunmall, B. (2003). Axioms and tests for the presence of minimal consciousness in agents. In O. Holland (Ed.), *Machine consciousness* (pp. 7–18). UK: Imprint Academic.

Aleksander, I., & Morton, H. (1993). *Neurons and symbols*. London: Chapman & Hall.

Baars, B. J. (1997). *In the theater of consciousness*. Oxford: Oxford University Press.

Barsalou, L. W. (1999). Perceptual symbol systems. *Behavioral and Brain Sciences, 22*, 577–660.

Cotterill, R. M. J. (2003). CyberChild: A simulation test-bed for consciousness studies. In O. Holland (Ed.), *Machine consciousness* (pp. 31–45). UK: Imprint Academic.

Davis, D. N. (2000). Minds have personalities – Emotion is the core. In *Proceedings of the AISB'00 Symposium on how to design a functioning mind* (pp. 38–46). U.K.: University of Birmingham.

Dennett, D. C. (1991). *Consciousness explained*. Boston, MA: Little, Brown and Company.

Duch, W. (1994, April). Towards artificial minds. In *Proceedings of the First National Conference on Neural Networks and Applications* (pp. 17–28). Kule.

Dudley, H. (1939). The vocoder, *Bell Labs Rec., 17*, 122–126.

Franklin, S. (2000). A "consciousness" based architecture for a functioning mind. In *Proceedings of the AISB'00 Symposium on how to design a functioning mind* (pp. 72–76). U.K.: University of Birmingham.

Franklin, S. (2003). IDA: A conscious artifact? In O. Holland (Ed.), *Machine consciousness* (pp. 47–66). UK: Imprint Academic.

Gibson, J. J. (1966). *The senses considered as perceptual systems*. Boston, MA: Houghton Mifflin.

Haikonen, P. O. (1994). Towards associative non-algorithmic neural networks. In *Proceedings of the IEEE International Conference on Neural Networks ICNN'94 Vol II* (pp. 746–750). Piscataway, NJ: IEEE.

Haikonen, P. O. (1998a). Assessor, a machine with functional consciousness. In *Consciousness research abstracts: Towards a science of consciousness, Tucson III* 117. UK: Imprint Academic.

Haikonen, P. O. (1998b). Machine cognition via associative neural networks. In *Proceedings of EANN'98* (pp. 350–357). Finland: Systeemitekniikan Seura ry.

Haikonen, P. O. (1998c). An associative neural model for a cognitive system. In *Proceedings of the International ICSC/IFAC Symposium on Neural Computation NC'98* (pp. 983–988). Canada: ICSC Academic Press.

Haikonen, P. O. (1999). *An artificial cognitive neural system based on a novel neuron structure and a reentrant modular architecture with implications to machine consciousness*. Dissertation for the degree of Doctor of Technology, Helsinki University of Technology, Applied Electronics Laboratory, Series B: Research Reports B4.

Haikonen, P. O. (2000a). An artificial mind via cognitive modular neural architecture. In *Proceedings of the AISB'00 Symposium on how to design a functioning mind* (pp. 85–92). U.K.: University of Birmingham.

Haikonen, P. O. (2000b). A modular neural system for machine cognition. In *Proceedings of the IEEE–INNS–ENNS International Joint Conference on Neural Networks IJCNN 2000* (Vol. I, pp. 47–50). Los Alamitos, CA: IEEE Computer Society.

Haikonen, P. O. (2002). Emotional significance in machine perception and cognition. In *Proceedings of the Second IASTED International Conference Artificial Intelligence and Applications* (pp. 12–16). Anaheim: ACTA Press.

Haikonen, P. O. (2003). *The cognitive approach to conscious machines*. UK: Imprint Academic.

Hinton, G. E., McClelland, J. L., & Rumelhart, D. E. (1990). Distributed representations. In M. A. Boden (Ed.), *The philosophy of artificial intelligence* (pp. 248–280). Oxford: Oxford University Press.

Holland, O., & Goodman, R. (2003). Robots with internal models: A route to machine consciousness? In O. Holland (Ed.), *Machine consciousness* (pp. 77–109). UK: Imprint Academic.

LeDoux, J. (1996). *The emotional brain*. New York: Simon & Schuster.

Plutchik, R. (1980). *Emotion: A psychoevolutionary synthesis*. New York: Harper & Row.

Rantala, A., & Haikonen, P. O. (2002). An associative neuron group microchip. In *Proceedings of the 20th Norchip Conference* (pp. 335–340). Copenhagen: Technoconsult.

Schachter, S., & Singer, J. E. (1962). Cognitive, social and physiological determinants of emotional state. *Psychological Review, 69*, 379–399.

Sloman, A. (2000). Introduction: Models of models of mind. In *Proceedings of the AISB'00 Symposium on how to design a functioning mind* (pp. 1–9). U.K.: University of Birmingham.

Taylor, J. G. (1992). Towards a neural network model of the mind. *Neural Network World, 6*, (92), 797–812.

Taylor, J. G. (1997, October). Neural networks for consciousness. *Neural Networks, 10*(7), 1207–1225.

Taylor, J. G. (1999). *The race for consciousness.* London: A Bradford Book, The MIT Press.

Trehub, A. (1991). *The cognitive brain*. Cambridge, MA: MIT Press.

Valiant, L. G. (1994). *Circuits of the mind*. Oxford: Oxford University Press.

Chapter 13

Does a Functioning Mind Need a Functioning Body? Some Perspectives from Postclassical Computation

Colin G. Johnson
University of Kent, UK

Abstract

In recent years, the idea that somatic processes are intimately involved in actions traditionally considered to be purely mental has come to the fore. In particular, these arguments have revolved around the concept of somatic markers, *i.e., bodily states that are generated by mind and then reperceived and acted upon. This chapter considers the somatic marker hypothesis and related ideas from the point of view of postclassical computation, i.e., the view that computing can be seen as a property of things-in-the-world rather than of an abstract class of mathematical machines. From this perspective, a number of ideas are discussed: the idea of somatic markers extending into the environment, an analogy with hardware interlocks in complex computer-driven systems, and connections with the idea of "just-do-it" computation.*

Introduction

One of the main topics of this book is the computational requirements for the existence of a functioning mind. This could either be a "purely rational" mind, or it could be a mind with affective capacity. In this chapter, I would like to consider to what extent it is possible for such a mind to exist in isolation from some form of "body." In particular, this question will be considered from the point of view of postclassical computation, which attempts to ground computation not in mathematical theories of abstract machines but by an analysis of the computational capabilities of the real world.

We will take a nondualist perspective as an axiom. Therefore, there is a requirement for the mind to be realized in some fashion in the physical world. The aim here is to consider the relationship between those parts of the body that act as a substrate for mind (in the sense that they could be replaced in a functionalist fashion by another substrate with no difference) and those parts of the body that influence mind yet that cannot/are not part of a substitutable substrate. The "cannot/are not" in the previous sentence can be interpreted usefully at a number of levels. A strong notion may be that there are no physically possible ways of realizing the same phenomenon. Some phenomena may admit a weaker notion in that it is "easier" in some sense (for example, faster, more energy efficient) for the mind to process this phenomenon using an alternative process rather than processing it on the neural substrate.

This chapter is structured as follows. The first section consists of a review that gives a context for the current work. This consists of an outline of notions of embodiment from the cognitive science and robotics literature and the core ideas of postclassical computation. The main sections of the chapter are concerned with in-the-world extensions to Damasio's theory of somatic markers (1994), connections between notions of the embodied mind and ideas of hardware interlocks in computing systems, and connections between the mind–body relationship and "just do it" computation. A final conclusion summarizes the arguments of the chapter.

Background

This section reviews the two main ideas on which this chapter is based. In the first part, notions of embodiment from the robotics literature are discussed. The second part discusses the idea of postclassical computation, i.e., computation that is based on the properties of physical objects in the world rather than on specific abstract models of computing machines. The final part looks at ways in which these two areas can be linked.

Notions of Embodiment

In recent years, much has been written about the importance of embodiment in the study of robots and autonomous machines. A definition of embodiment is given by Quick et al. (1999): basically, the embodied object is able to perturb some states of the environment and vice versa. There are a number of ways in which such notions are important for cognitive science research (Quick et al., 1999; Wilson, 2002; Ziemke, 2001, 2003).

In order for cognition to occur in the world, it has to be realized in some worldly "stuff." This alone is sufficient reason for considering notions of embodiment to be important to the arguments in this chapter. However, there is a second reason that is yet more relevant: the stuff in which cognition is realized will influence the cognitive capacity of the system. The brain-substrate, on which neural-network models of mind reside, opens certain pathways for, and places certain constraints on, the kinds of computations that can be carried out. By delegating parts of cognition to the body, a different set of computational affordances (Gibson, 1977; Norman, 1988) is provided.

A third perspective is that some procedures make use of their being-in-the-world as part of their functioning. Brooks (1991) said that "the world is its own best model," and many artificial cognitive systems consist of a reactive model, taking information from the world in an implicit fashion and reacting to that information, rather than building up an internal model of the world.

Another aspect of embodiment is that the actions of an embodied cognitive system are potentially unlimited. An action carried out on the mental substrate is limited by the computational capacity of that substrate. However, once a computation is sent off-substrate into the world, this restriction is removed.

Postclassical Computation

The aim of this chapter is largely to explore how the perspective offered by postclassical computation (a.k.a., nonclassical, nonstandard, physically grounded computation) can inform issues concerned with the embodiment of mind. Broadly speaking, postclassical computation is concerned with ideas of computation that come from the world, for example, studying the information-processing capabilities or the information storage capacity of some physical, chemical, or biological system. This contrasts with the (equally valid) "classical" view of computing as a property of certain classes of abstract machines. Readers should see Stepney et al. (2003) and Stonier (1990) for overviews of some of the ideas in this area.

The standard view of computing is that it is not concerned with things in the natural world. Grundy (1998) compared the existence of a science of computers to "a science called motorcarology." The gist of this argument is that computing is a synthetic subject. It is concerned with discovering (empirically grounded, theoretically supported) guidelines for the creation of computing machines. This contrasts with the analytic perspective on the world taken by a subject such as physics or psychology. In such a subject, the aim is to analyze things in the world using the tools and perspectives offered by the subject in question.

Most of the mature sciences have both an analytic side and a synthetic side to them. For example, the subjects of chemistry (analytic) and chemical engineering (synthetic) support each other in understanding and enabling the construction of objects at the molecular level. However, within computer science, there is traditionally no analytic perspective. One of the key approaches in postclassical computation is to consider a computational stance toward objects in the world. We can ask computational questions, such as the following: What is the information-processing capacity of that object? How much information can an object store? How quickly can a particular object transform information, and how does that contrast with how quickly a traditional computing device might do it? How is information compressed within that object? How does it achieve a certain kind of parallel processing of information? What constraints are placed on the object's function by the requirement to calculate certain characteristics, retrieve certain pieces of information, etc.?

It might be the case that computation has only accidentally been discovered through the creation of computing machines. Feasibly, computational notions could have been discovered by the analysis of certain objects in the world, before being applied in a synthetic fashion.

There are a number of areas in which we might see this kind of thinking being applied in the near future, or where we are beginning to see evidence of this approach already. One area is in immunology—the immune system can be viewed as a learning system that takes information from various invasions into the body and stores information about these in a network that helps the body distinguish self from nonself (deBoer et al., 1993).

This perspective on immunology allows us to ask computational questions about the immune system. What is the memory capacity of the system? How does the system retrieve information about previous infections quick enough to respond to reinfections by the same or similar antigen? How can the system recognize a wide variety of infections, and yet still have the capacity to swamp a particular infection with large amounts of specific lymphocytes when needed? Is information passed between lymphocytes to enable them to deal with infection, or is the apparent change in specificity a property of the population changing in composition? How quickly does a rapidly mutating antigen such as the HIV virus have to change to be able to change quicker than the rate of change in the immunological memory of the system?

Other areas of physiology can also benefit from being viewed in a computational fashion. Paton (1996) discussed the benefits to be gained from viewing certain processes in tissues as being a parallel and distributed system that processes molecular information. As an example, he discussed the processes found in the lining tissues of glands in terms of information processing. Using this, he was able to formulate testable hypotheses about why certain structures in glands have the forms that they do.

An interesting question is the extent to which computational capabilities of systems are able to influence the development of those systems. It has already been demonstrated that the dynamics of a system can influence the overall behavior of the system. An example comes from the study of certain blood diseases (Haurie et al., 1998). The distinction between two forms of a disease comes not from radically different concentrations of substances in the blood or from a large error in the system, but instead from small changes that upset periodic feedback patterns in the concentration of platelets in the blood, which then become very hard to return to equilibrium. Could there be similar problems that are best understood in terms of computational constraints?

Reasoning such as this can also be applied to physics. An example from cosmology is the work of Schmidhuber (1997, 2000) who has argued that it may make sense to choose between different "theories of everything" on grounds of computational complexity.

We demonstrated a number of different ways in which a computational attitude toward the material being studied allows us to ask and answer questions that are distinct from traditional scientific questions. This has been shown to be a valuable perspective in zoology, molecular biology, medicine, psychology, and physics. Answers to questions about the existence of structures in worldly phenomena that store and process information, and their information capacity and computational speed of action, provide valuable scientific information. Without looking at the world with computational eyes, these questions are unlikely to arise.

It is important to note that what we are doing here is different than simply simulating these systems on the computer. In traditional modelling, the questions being asked are identical to those that would be asked if the methodology being used was a conventional experimental one. In contrast, in these arguments, the nature of the model influences the kinds of questions being asked. Another difference is that in this approach, we can learn a lot from when the simulation "goes wrong" and does not reproduce the behavior seen in the real world.

Another important consequence is that this opens the possibility of exploiting the computational capabilities of the world to create new kinds of computation. For example, Feynman (1982) argued that we could respond in two ways to the "problem" that quantum systems are difficult to simulate on computers: we can regard this as a limitation on what computers can do, or we can regard this as an opportunity to build new kinds of computers that are grounded in quantum mechanical concepts.

A generalization of this argument is as follows. If we are faced with a problem in the world that appears to be inherently complex to compute, then we can take one of two stances. First, we can say that the process is "not computable," "difficult to compute," "computationally complex," etc. However, a second possibility arises—we can use that process as the basis of a new form of computation. One of the main points of this chapter is that such an exploitation may already have happened during the course of evolution of the mind, i.e., that the mind has learned to exploit certain somatic systems to do computational processes that might be too slow to carry out if done using on-substrate processes.

Combining these Perspectives

The main aim of this chapter is to consider some of the consequences of postclassical computation for the problems of embodiment. In particular, we can ask the question of when the embodied mind might make use of modes of computation that are not part of the substrate on which the (classically) "computational" parts of the mind are being implemented. These processes could be termed "off-substrate" computations. There are a number of reasons why we might expect such phenomena to be observed. The most radical is that there is some aspect of mental functioning that cannot in any way be carried out using conventional computing machines. Arguments of this kind have been made by Penrose (1989, 1995), who proposed that certain aspects of mental processing cannot be carried out in a (classical) computational fashion. Instead, he proposed that these functions might be carried out by quantum processes working alongside traditional notions of mental functioning.

However, there are a number of other levels of arguments that are weaker than those that show there are mental phenomena that are not computable in the classical sense. One important reason may be that the result is computable, but the use of some alternative process may facilitate faster computation. For example, it was shown by Adamatzky (2001) that a diffusive chemical can solve the problem of finding the shortest route through a maze. The chemical is released through the maze and takes all possible routes. When it reaches a branch-point, some of the chemical will diffuse in one direction, some in a different direction. This provides a kind of "parallelism on demand" that vastly improves the efficiency of the search contrasted to traditional search. It is feasible that off-substrate processing within the body is carried out for this reason.

Importantly, the potential presence of such off-substrate processes is supplementary to traditional on-substrate computation. One of the ways in which such computation may work is that the traditional on-substrate processing prepares information for input into an off-substrate process, and the output from that process is then further processed using conventional neural processes.

An additional reason for using processes other than conventional computation is the additional input/output capacities that are created by such processes. One feature of the neural-network brain that is typically advantageous is its capacity for processing many aspects of the world in parallel. However, this occasionally is to its disadvantage, for example, when the mind needs to "pull together" to dedicate most of its resources to attending to a danger signal. In such cases, there is no on-substrate mechanism to force the attention of various mind mechanisms toward the danger. By triggering a body state that will require attention by many different mental mechanisms (for example, a sudden feeling of nausea or a rapid heartbeat), there is a "massive synchronization" of mental resources, which provides a counterweight to the usual "massive parallelism" of mental functioning. This will be discussed below in the context of somatic markers.

Extended Somatic Markers

Damasio (1994) introduced the notion of the somatic marker. Somatic markers are bodily states that play a role in cognition, in particular, in the direction of attention. Specifically, a somatic marker is some bodily state that is generated as the consequence of some mental process. This state is then reperceived by the mind, and as a consequence, the mental state is changed. An example of such a marker is the rapid onset of nausea upon witnessing an act of violence. This bodily state does not have any immediate relevance to the mental state that has generated it, in contrast, say, to a feeling of nausea generated by viewing a plate of rotting food. Some such states might be explained away as side effects, for example, a rapid change of hormone levels upon witnessing violence in preparation for running from the danger might also trigger nausea.

The somatic marker hypothesis suggests that such reactions are not mere side effects. Instead, they are ways of generating rapid shifts of attention, using the body state in an arbitrary fashion to draw mental attention to the current situation.

The presence of the marker in the body draws the mind's attention toward it, and as a consequence, the mind is focused on the meaning of that marker. It is plausible that such phenomena are exaptations [i.e., co-options of previously evolved functions to new ends (Gould & Lewontin, 1979; Gould, 1991)] from unwanted physical reactions to changes in body state, as discussed above.

We can see this as an example of off-substrate computation. The somatic response is being used as a way of carrying out a process (bringing the attention of many mental processes together to focus on a single danger point) that cannot be carried out within the computational model implemented on the substrate.

An interesting question is whether it is important that such markers be somatic, i.e., need they be internal body states? There are two questions to be answered here. First, we can consider why it is important for the marker to exist in the body and not just in the mind. Reasons for this are detailed in Damasio's book, and a particular viewpoint on this is given in the next section of the chapter. This section will focus on the contrasting question: why does the marker need to be constrained to reside within the body? One approach to this draws on ideas from Dawkins' (1992) book, *The Extended Phenotype*.

In biology, the phenotype is the expression of a gene or set of genes in the world, for example, through physical structure or through influences on behavior. For example, we can talk about the "blue-eyed" phenotype versus the "brown-eyed phenotype" of some animal. This is distinguished from the "genotype," i.e., the set of genes of interest. Sometimes, more than one genotype can give rise to the same phenotype (for example, where there are regressive traits).

The difficulty starts when we want to say where the boundary of the phenotype lies. Clearly, certain things are in the phenotype, for example, the sequence of proteins associated with a particular expression of a particular gene. A standard definition would extend this to the whole body—genes influence the growth, development, and activity of the body (alongside other influences).

Dawkins' argument is that it is naive to simply say "everything inside the body, phenotype; everything outside, not." As an example, consider an imaginary species of bird in which the male has a gene that predisposes itself to mate with females that have blue feathers; it could be said that this gene is also a gene for blue feathers in the female, and as a result of the presence of the gene, blue feathers will spread through the female population. To abstract this, the genotype in the male bird is having a phenotypic effect in the female bird. Why should we regard the gene's effect on the feathers of the female bird in any different way than we regard another gene that causes the male bird to have red eyes?

A similar kind of argument can be made about the somatic marker hypothesis. Damasio argued for a body-minded brain in which we create emotions via "somatic markers." These work when parts of the brain recognize an emotionally charged stimulus, and rather than create a direct link to an action on that stimulus, the "marker," consisting of a bodily reaction, is created. This is then reperceived by the brain as the basis for action or for rapid alteration of emotional state. Why do these markers have to be physically internal to the body? It would seem that the same reasoning could be applied to markers that I leave in the external world when I have an emotion. For example, if I am anxious, then I might scribble on the pad of paper in front of me, without attending to this scribbling. This could then become a marker, in this case, perceived via the eyes rather than through proprioception. Why should it matter whether I use a bodily state or an external state as the substrate for the marker?

It may be that there are reasons why somatic markers need to be somatic. One could be that the speed of reaction required is just too quick to be capable of being carried out by the external perceptive system. Another more convincing explanation is that the reason we use somatic markers is to communicate with multiple brain regions in a simultaneous and coordinated way, and therefore, we need something that can be perceived in a direct way by different parts of the brain.

This might be a continuum effect. An example of a phenomenon that might be seen as either an external or somatic marker is biting nails when anxious. This is, in many ways, an external physical process, but, nonetheless, we can perceive the nail state internally via soreness of fingers. There must be other similar examples. Perhaps nail-chewing is "causing" the anxiety (in the sense of being part of the causal chain between subconscious perception of an anxiety-producing stimulus and the affective response) rather than being an epiphenomenon of the emotional state.

Hardware Interlocks

In the previous section, we asked why the somatic marker needs to be constrained to the body, and whether it is important to make a body-nonbody distinction. In this section, we address the opposite question: why is it not sufficient for the marker to be a mental marker? Why not just make a "mental note"? While there are circumstances in which a truly somatic marker can get transformed into a mental process in the limbic system, it is interesting to consider whether there might be reasons why the evolution of the mind might have led to the markers being body-centered rather than mind-centered.

One reason may be for safety. In the design of complex systems involving computer-controlled mechanical and electrical devices, it is common for there to be conservative safety devices included in the system, known as hardware interlocks (Leveson & Turner, 1993; Leveson, 1995). A hardware interlock is a device that is independent of the main control system and that is designed to monitor one small aspect of the system, typically by using its own sensor system. For example, in a radiotherapy device, an interlock might exist that monitors the output of radiation, and if more than a certain amount is let out in 1 minute, the interlock shuts down the device completely.

Hardware interlocks are designed to be parts of the overall system that do not depend on the abstraction offered by the overall control system. For example, they do not take information from the main system sensors, they do not use the main control system (for example, for timing), and they do not sit upon the operating system abstraction used by the controlling structure. To do this would compromise their role as safety-critical components. They provide a reassurance of safety because they are separate; they are independent from the main abstraction. If the main sensors go wrong, or the builder of the controller has misunderstood the relationship between the abstraction offered by the operating system and the real hardware and software, it does not matter.

One important role of the body-mind system is to react quickly and reliably to dangerous phenomena. There would seem to be a *prima facie* case for thinking that if engineers consider the use of such hardware interlocks as an important way of responding to danger in computer-controlled systems, evolution may have created such interlock systems for dangers to animals. It may be that our body-grounded response to danger is a response of this kind. Instead of making a mind-centered judgement about the danger of a situation, we make a rapid decision based on a few simple cues. One characteristic of hardware interlocks is that they typically work on a small number of basic sensors that facilitate a conservative approximation to safety. The same may be true of interlocks in the mind-body system: our sensory system perceives a small number of simple "danger signals" (such as a rapid movement) and triggers an action within the body immediately. This "massive synchronization" acts as a counterpart to the more commonly discussed "massive parallelism" of the neural-network-based mind.

Typically, the fact that the brain is a unified system with all aspects connected and mutually accessible is seen to be to its advantage. Similarly, the unity found in a complex software system is often seen as being to its advantage; instead of having to connect individual components as needed (as might be the case in an electronic system), all information is passed to a central repository and accessed as needed. In some situations, it is necessary, for computers and for minds, for the complete attention of the system to be directed toward one thing. Hardware interlocks provide a way for such responses to "leap out" of the complexity of the control software for certain emergency situations. This nondecomposability, and the consequent need for a powerful way of leaping out of the complex interactions, would seem to be particularly strong for neural-network-based systems, where the system is highly nondecomposable.

"Just Do It" Computation

Traditional computing is concerned with the construction of machines based on various mathematical models of computing machines. One of the ideas in postclassical computing is that many kinds of transformations in the world can be regarded as computations by consistently ascribing informational values to the objects involved in those transformations. Such "computers" carry out their computations without regard to traditional notions of computational complexity.

For example, proteins fold consistently into complex three-dimensional shapes, despite the complexity of this process. It was shown that a simplified model of the protein-folding problem is NP complete (Berger & Leighton, 1995; Crescenzi, 1998; Fraenkel, 1993), and that an exhaustive search of all configurations would take on the order of 10^{45} years (Levinthal, 1969). Nonetheless, real proteins fold reliably within seconds. Similarly, adding new stars to a galaxy does not slow it for computational reasons.

In such processes, the information (the positions of things in the world) is transformed without any explicit computational effort; the object "just does it." We can imagine a new kind of computer-based problem solving based on this. In traditional computation, the role of the computer is to compute the solution to a problem directly, by applying algorithms that transform some representation of the input into some representation of the output. However, if a sufficiently flexible set of "just do it" (JDI) devices can be brought together, then there remains the possibility of a new kind of computation. Instead of building a single computing device, computational problem solving is seen as being about the preparation of input for JDI devices, which are then allowed to complete their calculations, output read offs, and interpret them. More complex processes may require a number of JDI processes to be carried out.

A toy example of this is given by Dewdney (1988). This is an *O(n)* sorting algorithm. A number of lengths of (uncooked) spaghetti are cut to the lengths of the input to the algorithm. These lengths are gathered together and stood upright on the table. The lengths can then be read off one-by-one from the longest downwards. Note that the complexity of the process increases linearly with the number of items being sorted. By contrast, a traditional computer increases in complexity by *O(n*log*n)* while carrying out a sorting process. The process of standing the spaghetti on the table is a massively parallel JDI process.

It is possible that some of the off-substrate processes in the somatically extended mind are of this type, i.e., the on-substrate computation is preparing the input for certain somatic processes to compute. In particular, decision making may be of this kind. The "rational" outcomes of various options are being computed on-substrate, and the final decision is made by a "gut feeling." A related argument has been made by Evans (2002). His argument is that one of the roles of emotion is to solve the "search problem," i.e., the problem of knowing where to stop when calculating the consequences of a decision, that emotions "prevent us from getting lost in endless explorations of potentially infinite search spaces."

These "just do it" processes are similar to what Kauffman (1993, 1995) described as "order for free." For example, in this viewpoint, the occurrence of the Fibbonacci sequence in the phyllotaxis of pinecones is not some miracle of nature; it is simply the energetically cheapest way of generating the desired structure.

A main part of his argument is that evolution will exploit such "order for free" as an energetically cheap way of constructing complex organisms. Instead of constructing new devices to achieve action, evolution pieces together various devices available in the world to construct complex biological machines. In this section, we made a similar argument with regard to the evolving mind exploiting available computational devices in the world.

Conclusion

Does a functioning mind need a functioning body? Perhaps yes, if some parts of mental functioning are delegated to off-substrate processes, either for the purposes of computational efficiency or because certain processes (such as the "massive synchronization" required to respond efficiently to danger) are not available in the on-substrate model.

One perspective from which to view this is that of postclassical computation. From this point of view, computation is seen as a property of things in the world, rather than only as a property of specially constructed computing machines. This allows us to ask questions about the computational capabilities of many different objects. In particular, we can ask questions about the computational capabilities of on-substrate processes and compare these to off-substrate processes. In some cases, the off-substrate processes may have properties that are not available on-substrate or that operate in a more efficient fashion than their on-substrate equivalents. In such cases, we have a *prima facie* case for considering that the off-substrate way of realizing that process might have been favored during the evolution of mind.

A particular example of an off-substrate process is the somatic marker, where a body state is used as shorthand for some mental state that needs to be rapidly appreciated by a number of parallel mental processes. This postclassical computation stance allows us to consider the relationship between these on- and off-substrate processes, and it provides the beginning of a way of determining whether particular processes are likely to be carried out on-substrate using connectionist networks, or whether they are likely to be delegated to other parts of the body.

References

Adamatzky, A. (2001). *Computing in nonlinear media and automata collectives*. Bristol: Institute of Physics Publishing.

Berger, B., & Leighton, T. (1998). Protein folding in the hydrophilic–hydrophobic (HP) model is NP-complete. *Journal of Computational Biology, 5*(1), 27–40.

Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence, 47*, 139–159.

Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., & Yannakakis, M. (1998). On the complexity of protein folding. *Journal of Computational Biology, 5*, 423–465.

Damasio, A. (1994). *Descartes' error: Emotion, reason and the human brain*. New York: Gosset/Putnam Press.

Dawkins, R. (1982). *The extended phenotype*. Oxford: Oxford University Press.

de Boer, R. J., van der Laan, J. D., & Hogeweg, P. (1993). Randomness and pattern scale in the immune network: A cellular automata approach. In W. Stein & F. J. Varela (Eds.), *Thinking about biology*, Santa Fe Institute Studies in the Sciences of Complexity: Lecture Notes (Vol. III, pp. 231–252). Reading, MA: Addison-Wesley.

Dewdney, A. K. (1988). *The armchair universe: an exploration of computer worlds*. New York: Freeman.

Evans, D. (2002). The search hypothesis of emotion. *British Journal for the Philosophy of Science*, 53, 497–509.

Feynman, R. P. (1982). Simulating physics with computers. *International Journal of Theoretical Physics, 21*, 467–488.

Fraenkel, A. (1993). Complexity of protein folding. *Bulletin of Mathematical Biology, 55*(6), 1199–1210.

Gibson, J. J. (1977). The theory of affordances. In R. E. Shaw & J. Bransford (Eds.), *Perceiving, acting and knowing*. Mahweh, NJ: Lawrence Erlbaum.

Gould, S., & Lewontin, R. (1979). The spandrels of San Marco and the Panglossian paradigm: A critique of the adaptationist program. *Proceedings of the Royal Society of London, Series B, 205*(1161), 581–598.

Gould, S. J. (1991). Exaptation: A crucial tool for evolutionary psychology. *Journal of Social Issues, 47*, 43–65.

Grundy, F. (1998). Computer engineering: Engineering what? *AISB Quarterly, 100*, 24–31.

Haurie, C., Dale, D. C., & Mackey, M. C. (1998). Cyclical neutropenia and other periodic haematological disorders: A review of mechanisms and mathematical models. *Blood, 92*(8), 2629–2640.

Johnson, C. G. (2003). What kinds of processes can be regarded as computations? In R. Paton, H. Bolouri, M. Holcombe, J. H. Parish, & R. Tateson (Eds.), *Computing in cells and tissues: Perspectives and tools of thought*. Heidelberg: Springer.

Kauffman, S. (1993). *The origins of order.* Oxford: Oxford University Press.

Kauffman, S. (1995). *At home in the universe*. New York: Penguin.

Leveson, N. (1995). *Safeware: System safety and computers*. Reading, MA: Addison-Wesley.

Leveson, N., & Turner, C. (1993). An investigation of the Therac-25 accidents. *IEEE Computer, 26*(7), 18–41.

Levinthal, C. (1969). How to fold graciously. In J. T. P. DeBrunner & E. Munck (Eds.), *Mossbauer spectroscopy in biological systems* (pp. 22–24). Illinois: University of Illinois Press.

Norman, D. A. (1998). *The psychology of everyday things*. New York: Basic Books.

Paton, R. (1996). Metaphors, models and bioinformation. *BioSystems, 38*, 155–162.

Penrose, R. (1989). *The emperor's new mind: Concerning computers, minds, and the laws of physics.* Oxford: Oxford University Press.

Penrose, R. (1995). *Shadows of the mind.* New York: Vintage.

Quick, T., Dautenhahn, K., Nehaniv, C. L., & Roberts, G. (1999). On bots and bacteria: Ontology independent embodiment. In *Proceedings of the Fifth European Conference on Artificial Life.*

Schmidhuber, J. (1997). A computer scientist's view of life, the universe, and everything. In C. Freksa, M. Jantzen, & R. Valk (Eds.), *Foundations of computer science: Potential — theory — cognition* [Lecture Notes in Computer Science] (pp. 201–208). Heidelberg: Springer.

Schmidhuber, J. (2000). *Algorithmic theories of everything.* [Technical Report 20-00.] IDSIA.

Stepney, S., Clark, J., Tyrell, A., Johnson, C., Timmis, J., Partridge, D., Adamatsky, A., & Smith, R. (2003). *Journeys in non-classical computation.* National E-Science Centre Grand Challenge report.

Stonier, T. (1990). *Information and the internal structure of the universe.* Heidelberg: Springer.

Wilson, M. (2002). Six views of embodied cognition. *Psychological Bulletin and Review, 9*(4), 625–636.

Ziemke, T. (2001). Are robots embodied? In C. Balkenius, J. Zlatev, C. Breazeal, K. Dautenhahn, & H. Kozima (Eds.), *Proceedings of the First International Workshop on Epigenetic Robotics: Modeling Cognitive Development in Robotic Systems.* Lund University Cognitive Studies (Vol. 85). Lund, Sweden.

Ziemke, T. (2003). What's that thing called embodiment? In *Proceedings of the 25th Annual Meeting of the Cognitive Science Society*. Mahweh, NJ: Lawrence Erlbaum.

Chapter 14

APOC: An Architecture Framework for Complex Agents

Matthias Scheutz
University of Notre Dame, USA

Abstract

In this chapter, we introduce an architecture framework called APOC (Activating-Processing-Observing-Components) for the analysis, evaluation, and design of complex agents. APOC provides a unified framework for the specification of agent architectures at different levels of abstraction. As such, it permits intermediary levels of architectural specification between high-level functional descriptions and low-level mechanistic descriptions that can be used to connect these two levels in a systematic way.

Functional Organization and Agent Architectures

Research on agent architectures has come to play an increasingly important role in the design of control mechanisms for complex artificial agents in the field of artificial intelligence (AI). It is predated by efforts in other related fields (in particular, cognitive psychology, ethology, or philosophy), where processing architectures have been proposed for living creatures (typically under different names, such as "cognitive architecture," "behavioral architecture," or "functional architecture"). Common to all these architectural approaches is the claim that the functional organization of agent control systems is the right level of abstraction at which to understand what brings about the behaviors of agents. Furthermore, it is this level that underwrites causal explanations of behaviors and warrants talk about mental states, such as "believing that *p*" or "desiring *x*" (including mental causation such as that my believing that *p* and desiring *x* caused me to act in a certain way, see also Kim, 1996).

The general picture here is that mental states are functional states, which, in turn, are realized in physical systems. In biological organisms, these physical states are brain states, but there may be other physical substrates that can serve as realizers (states of computers, for example). However, there are currently no formal accounts of what functional architectures are, or what they would look like for a complex, humanlike agent, although there are some preliminary high-level proposals, for example, Sloman (2000). And even if there were such accounts in terms of causal networks of high-level functional states, they would likely not be very helpful to AI researchers who intend to build complex agents based on such architectures, as too many details would be missing to point to possible implementations. In addition, a series of problems would likely result from the hitherto ill-defined notions of "functional realization and implementation" (Scheutz, 1999, 2001a).

To see that, consider the implementation conditions for a functional architecture. To say that a system implements a functionalist description is to require that in addition to the input and output mapping, it has to get the mapping of the "inner states" right, which are assumed to be multiply realizable (i.e., many different, possibly diverse physical systems will realize a given functional architecture). Therefore, the mapping between physical states and functional states has to be a many-to-one (in the spirit of Chalmers, 1997). Yet, inner states are viewed by functionalists as intrinsically relational states, being mutually defined by all states in the functional architecture (which is often expressed by saying that they are defined by their causal role in the functional architecture). Because functionalists only require that there be an arrangement of physical states that corresponds to the functional states in a way that preserves inputs and outputs as well as transitions between states, it is possible for one physical state to serve as the instantiation of more than one functional state (and vice versa). Therefore, the correspondence between physical and functional states is not necessarily that of a mapping between physical types and functional types (let alone a 1-1 mapping), but rather that of a relation preserving state transitions (i.e., some sort of bisimulation[1]).

Even if we had a clear account of functional realization, and it were possible, using this account, to relate high-level functional states directly to low-level physical states, this relation would unlikely give us any insight into the inner workings of a complex system. For one, functional states are typically holistic, and their inner structures are essentially obscured. What is more, the causal mechanisms that bring about the instantiation of these states cannot be understood in terms of these states, even if causal relationships are viewed as state transitions among them. Therefore, a framework is needed in which functional architectures of agents can be specified in a way that allows us to see their high-level functional organization, while permitting us to specify the architecture in terms of functional components that can actually be implemented (for example, in computers).

Interestingly, the need for such an architecture framework can be argued from a different perspective—that of practical agent design in artificial intelligence. In the course of the last three decades, a large number of architectures for intelligent agents has been proposed (partly based on different design methodologies), many of which were developed for particular kinds of agents and targeted at a particular class of tasks, from high-level cognitive architectures, such as SOAR (Laird, 1987) or ACT-R (Anderson, 1993); to logic-based reasoning and planning architectures, such as BDI (Rao & Georgeff, 1991) or STRIPS (Fikes & Nilson, 1971); to behavior-based architectures such as Subsumption (Brooks, 1996) or Motor Schema (Arkin, 1989); to real-time architectures for robots, such as 3T (Bonasso et al., 1997) or RCS (Albus, 1993). Because most architectures have proven successful in their application domains, it would be advantageous if it were possible to reuse components and principles that contributed to the success, particularly, if they could be utilized in other architectures that do not use the same basic components or design methodologies. Yet, this is problematic for at least two reasons: it may be difficult to say what part of an architecture or design accounts for its success, and it may be difficult to compare two different architecture types directly, because their design assumptions and domain restrictions may vary significantly (such as symbolic versus subsymbolic or nonsymbolic, high-level versus low-level, serial versus parallel, software versus robotic agents, etc.).

One way to assess the advantages and disadvantages of particular designs and methodologies and utilize them in other designs is via an agent architecture framework that is general enough to allow researchers to evaluate and compare different kinds of architectures. Because the framework provides a unified basis for expressing different architectural mechanisms, successful mechanisms can be, in principle, integrated within one architecture. Whether a combination of different mechanisms makes practical sense (and is functional) will depend on the given task.

Note that such an architecture framework needs to be conceptually parsimonious enough to employ only a few intuitive, basic concepts by virtue of which concepts in other architectures can be defined. Otherwise, the framework may end up being as complex as any of the higher, universal programming languages in which agent architectures are defined—obviously, such a framework would defeat its purpose. To our knowledge, no satisfactory framework is available yet.

As a step toward the development of such a general framework, we will introduce an architecture framework that we have developed over the last several years called APOC (Activating-Processing-Observing-Components) (Andronache & Scheutz, 2002, 2003a, 2003b, forthcoming; Scheutz & Andronache, 2004). APOC not only provides a unified framework for the specification of architectures of minds at different levels of abstraction, but it also provides an intermediary level of architectural specification between functionalist views (as favored by philosophers) and physical descriptions (as favored by neuroscientists) that connects these two levels in a systematic way and allows the AI researcher to design agent architectures to be implemented on computers. Being conceptually simple and small (in the number of concepts), while achieving high expressiveness at different levels of abstraction, it satisfies the above requirements and allows researchers to analyze, evaluate, and compare agent architectures in a unified way. To facilitate the design of complex agents, we created a practical tool based on APOC—the APOC development environment (ADE) in JAVA—which allows for the distributed implementation of architectures specified in APOC on multiple computers, and furthermore supports the interaction with simulated and robotic agents in a transparent way.

In the following, we first introduce the APOC framework and then discuss its potential for the analysis, comparisons, and design of agent architectures.

Overview of the Architecture Framework "APOC"

APOC consists of heterogeneous computational units, based on Scheutz (2000, 2001b), called components, that can be connected via four link types to form an agent architecture. The four link types cover important basic interaction types among components in agent architectures: the *activation link* allows components to communicate with other components; the *observation link* allows components to observe the states of other components; the *process control link* enables components to influence the computation taking place in other components; and finally, the *component link* allows a component to instantiate other components and connect to them via the other three links.

Components can vary with respect to their complexity and the levels of abstraction at which they are defined. They could be as simple as a connectionist unit or as complex as a full-fledged condition-action rule interpreter. APOC can be used as an analysis tool for the evaluation of architectures, because it can express any agent architecture in a unified way (cognitive architectures such as SOAR, ACT-R, and others, as well as behavior-based architectures, such as subsumption, motor schemas, situated automata, etc.).

APOC also supports the idea that mental states (or concepts) can be defined in terms of architectural capacities of agent architectures (Sloman, 2000, 2002). Hence, component and link structures in APOC can be used to define minimal requirements for the presence of mental states: an architecture *A* is capable of instantiating a mental state *S*, if the architecture-based definition of *S* in terms of APOC is a substructure of *A*. Thus, APOC is a first attempt to provide a framework that not only allows for a detailed computational specification of architecture-based concepts, but also provides an immediate implementation of the specification, thereby contributing to the resolution of one of the most critical questions in the foundations of AI: how do we know that a given physical system instantiates a given mental state? Systematic investigations of APOC structures can then be used to develop an architecture-based taxonomy of possible cognitive and affective states (see also, Sloman & Scheutz, 2002).

APOC also introduces a novel idea that is essential for the study of the computationally plausible theory of minds: the notion of *cost induced by an architecture*, which is defined in terms of the cost associated with *structures*, *processes*, and *actions on the architecture*. *Structural costs* are those that are incurred as a result of merely having a certain component or link instantiated. They can be thought of as maintenance costs that are associated with any work that needs to be done to keep a component or link up to date. *Process costs* are those associated with running processes; they include computational costs, and possibly the costs of input/output and other such operations. Typically, process costs will be proportional to the complexity of the computation performed by the process. Finally, *action costs* are those associated with primitive operations on the architecture (such as instantiating a new component or link, or interrupting a process). Each action has a fixed cost, making the computation of action costs a simple matter of assessing the associated cost whenever the action is executed. The notion of cost induced by an architecture is then inductively defined in terms of these three basic cost types.

Using the notion of cost induced by an architecture, the notion of *performance-cost trade-off* **PCT**(P,A,T,E) for an agent architecture A and a task T in an environment E can be defined as P/C, where P is the given performance measure for T, and C is the cost of A for T in E. Mathematically, performance-cost trade-offs are orders, and can thus form the basis of the comparison of agent architectures: given an order $>_p$ defined on P, an architecture A is said to be better than an architecture B with respect to T, E, and P, if $\mathbf{PCT}(P,A,T,E) >_p \mathbf{PCT}(P,B,T,E)$. Hence, APOC allows for a novel comparison of different agent architectures in terms of the cost induced by them. This will help in answering questions about evolutionary trajectories of architectures of biological organisms, as evolution will always favor architectures with higher performance-cost trade-offs.

We believe, although we will not be able to argue this here, that the notion of cost induced by an architecture is crucial for any theory of mind that takes the nature and constraints of real-world agents into account, for complex minds have to intrinsically cope with resources restrictions. In particular, we conjecture that (efficient) resource management is one dimension along which to evaluate designs of complex agents (evolution certainly did).

After this brief overview, we will now give a more detailed characterization of the APOC architecture framework.

Generic *APOC* Components

All five types of entities, APOC nodes and the four link types, are derived from one generic type, call it *generic APOC component*. A network of APOC nodes, which are connected to each other by virtue of APOC links, is called an *APOC architecture schema*. We use the term "architecture schema" here to emphasize that these networks of connected nodes may be "schematic," depending on the extent to which each node is schematic (i.e., left unspecified with respect to its exact functionality—see the following). Before we can specify the details of the five APOC component types, we first need to define the generic APOC component and its properties.

Generic APOC components are intended as general autonomous control units that are capable of updating their own state, influencing each other, and controlling an associated process. The exact nature of associated processes is not specified (they could be computational or merely physical processes), except that they can be in one of four different states at any given time: ready, running, interrupted, and finished. This is the smallest set of states that allows an APOC node to control its associated process without making any specific assumptions about the nature of the process (as would, for example, be the case if states like stalled or blocked were to be introduced).

If a component has a process associated with it, the component can control the process by virtue of three primitive actions: start will start the process if it is ready, interrupt will interrupt a running process, and resume will resume an interrupted process. Whether processes that are finished can and will become ready depends on the nature of the process (for example, whether the process can be restarted as in the case of a "socket listening process," which upon closing the socket finishes, but which can become ready again when restarted, or whether it is a terminal process, such as a "cleanup process" that shuts down the electric circuitry connected to an effector of a robot permanently, only to be restarted at the next system level restart).

Generic components have input and output ports that can be connected to output and input ports of other components, respectively. A set of connected components, then, forms a network of components, i.e., an architecture schema.

Every generic component *C* has an activation level that reflects (in part) the internal state of the component and can be influenced by inputs, previous states, and various operations performed either on or by the component (to be discussed below). Exactly how the state of a component changes is determined by its update function that maps inputs, internal states, and the state of the associated process to outputs, internal states, and operations on input and output ports as well as the component. More precisely, an update function *F* is a mapping $F\colon ST \times PST \times IN^{k} \rightarrow ST \times OUT \times OP^{3+k+l}$, where *ST* is the set of internal states of generic components, *PST = {ready, running, interrupted, finished, no-proc}* is the set of possible states of the associated process as mentioned above (*no-proc* indicates that no process is associated with the node), *IN* is the set of input states, *OUT* is the set of output states, and *OP* is the set of operations a component can perform. The set of operations can be subdivided into operations that a component can perform on its associated process and possibly the processes of other components *POP = {start, interrupt, resume, noop}*; those it can perform on itself and other components *per se COP = {instantiate, terminate, noop}*; and finally, those it can only perform to manipulate its priority *SOP = {incr, decr, noop}*, such that $OP = POP \cup COP \cup SOP$. Note that the *noop* is used formally if no operation is performed. All these sets together fix the set of possible architecture topologies of generic components as well as their operations and input and output state types.

Components also have a priority level and an instantiation number associated with them. The former is used to determine whether other components connected to *C* (through one of its ports) can influence the process associated with *C* (if *C* has an associated process), and whether *C* can influence the associated process of the components it is connected to through its ports. The latter is used to determine whether components connected to *C* (through one of its ports) can instantiate a component of type *C*, and whether *C* can instantiate components of the types it is connected to through its ports.

Components can control (i.e., *interrupt, resume*, and *start*) the processes associated with any component they are connected to through their input or output ports, if their current priority is higher than the priority of that component.

The controlled component cannot override the externally controlled operation unless it manages to change its current priority to at least the level of the external, controlling component. Components can change their priority levels one step at a time (as defined by the order in the set of priority levels) up to the maximum level. By the same token, they can also lower their priority levels to the lowest level. If multiple components attempt to control (the associated process of) a component, the component with the highest priority wins. If the highest component is not unique, and different operations are attempted by the highest components, none of their operations will be performed. This mechanism allows for the implementation of a great variety of arbitration schemes, from competitive winner-takes-all schemes to cooperative fusion schemes (for a detailed description of how common action-selection schemes can be implemented in APOC, see Scheutz and Andronache, 2004).

Components can only instantiate other components up to their own instantiation limits, i.e., every time the instantiate operation is performed, the current instantiation number is increased, up to the maximum instantiation number, at which point no more components can be instantiated (this number effectively fixes the number of output ports). Similarly, every time the terminate operation is performed, the current instantiation number is decreased. Components can only terminate other components if they have instantiated them. If the current instantiation number is one, and a component performs a terminate operation, it effectively terminates itself and ceases to exist.

The instantiation process in APOC is similar to a bootstrapping process of virtual machines in standard computers on power-up: one initial component is instantiated as specified in its initial state, which will then take care of instantiating all other components in subsequent stages.

Once generic components are instantiated in a virtual machine, they are self-sufficient entities that behave according to their specifications as determined by their initial states, their associated processes, and their update functions. To be able to define the state of a generic component, we need to fix three more sets: *ACT*—the set of activation levels, *PRO*—the set of processes (containing nonproc), and *PRI*—the set of priority levels. The state of a generic APOC component can be defined as the 8-tuple $<act,pri,pro,inst,F,in,out,op>$, where $act \in ACT$ is the activation level, $pri \in PRI \times PRI$ is a pair containing the current and the maximum priority levels, $pro \in PRO \times PST \times POP$ is a triple containing the process state and the process associated with node as well as the operation performed on that process, *inst Î* $N \times N$ is a pair containing the current (integer) instantiation number and the maximum number of instances of a node of that type, and $F \in UF$ is the update function.

Furthermore, in and out are k- and l-tuples, respectively, of triples $<m,p,n>$ that reflect the states of k input and l output ports of the node, where $m \in IN \cap OUT$

is the message received from or sent to port *p* of node *n* (a tuple of the form <*m*,*p*,∅> indicates that port *p* is not connected to any other node).

Finally, *op* is a (*3*+*k*+*l*)-tuple, where the first three operations concern the component itself (its process, its priority, and its instantiation capacity), the next *k* its input, and the last *l* its output ports. On each port, the component can either terminate a connected component (if it instantiated it previously) or manipulate that component's associated process. Other operations will not have any effect.

Note that the state of a generic node contains its update function, different from (finite) state automata, where the state transition function is not part of the state of an automaton. This is to allow nodes to change their update functions over time. Hence, states and update functions are defined mutually in terms of each other. While circular definitions resulting in non-well-founded sets are not allowed in standard set theory (largely because certain malicious definitions give rise to paradoxes), not all circular definitions are problematic (Barwise & Moss, 1996). Such unproblematic definitions can be formulated in non-well-founded set theory (Aczel, 1988), where it is possible to define structures (i.e., sets) that mutually contain each other using the *General Solution Lemma* (Barwise & Moss, 1996). For lack of space, we will not be able to provide any details here of how the APOC architectures can be defined using the *General Solution Lemma* in non-well-founded set theory.

APOC Nodes and Links

The five APOC types are obtained from generic APOC components by virtue of restricting the set of possible states and, hence, the set of permissible update functions. The idea is to provide a set of basic types that more closely resemble the kinds of components found in typical agent architecture, are generic enough to be able to define common architecture components at different levels of abstraction, are detailed enough to be (directly) implementable in higher programming languages, and are suitable to support causal explanations of agent behavior (i.e., of the effects of events in instantiated agent architectures).

APOC nodes are connected to other nodes through one of the four previously mentioned APOC links. Each node has associated with it a pair <*id*,*max*>, where *id* is the instantiation number of the node, and *max* is the maximum number of nodes of that type that can be instantiated in an APOC architecture. Note that the functionality of these numbers is different from that of generic nodes in that it makes reference to a "global instantiation maximum" rather than to local one. It can, however, be defined in terms of additional nodes and the local mechanisms of generic nodes (see the description of component links below).

Because the restrictions on the set of possible update functions of APOC nodes are determined by the functionality of these four links, it is sufficient to discuss the four link types, which are specialized generic components. Links have no processes associated with them and their input and output ports are usually restricted to APOC nodes, one for the input and one for the output port.[2] Moreover, there is typically only at most one link of a given type between any two APOC nodes. We will, therefore, restrict the formal definitions of links to links with two ports, where the components connected on the two ports are APOC nodes.

Activation links are the most general means by which nodes can exchange information. Their state is given by the tuple *<act,0,∅,inst,S,R,F>*, where *S* is the node providing the input, *R* is the node receiving the output, *act* the activation level of the link (typically dependent on the input provided by *S*), and *F* the update function. Because links do not have processes associated with them, their *pri* value is set to *0*. Furthermore, the maximum instantiation number of activation links, which is part of the pair *inst*, is given by the corresponding number in *inst* of *S*. The purpose of activation links is to connect two APOC nodes and serve as transducers. They can be used in a variety of different ways. In the simplest case, they function as mere connections between input and output ports of APOC nodes (i.e., inputs to links are identical to their outputs). It is also possible to implement a *timed link*, i.e., a delay, with which the value at the output port of *S* arrives at the input port of *R*. Furthermore, an activation link can be used transform the input. In case of numerical values, it could, for example, *scale* the input by a particular factor (analogous to the *weights* on connections in neural networks).

Priority links are intended to explicate the capacity of generic components to control other components' associated processes. They are the only means by which APOC nodes can control processes of other nodes (because no link has a process associated with it, and nodes can only be connected to other nodes via links, APOC nodes could not control any process otherwise). The state of a priority link is given by the tuple *<act,0,∅,inst,S,R,F>*, where *S* is the node attempting to take control of the process associated with *R*, and *F* is the identity function. The other parts are the same as with the activation link. A priority link effectively passes the process control request of an APOC node on to the node it is connected to. Priorities can be used to implement all kinds of control mechanisms, in particular, hierarchical preemptive process control. In embodied agents, such as robots, they could be used to implement emergency behaviors: the node with the associated emergency process would have the highest priority in the network and be connected to all the other nodes controlling the agent's behavior, which it could suppress in case of emergency, thus implementing a *global alarm mechanism* as described in Sloman (1998).

Observer links are intended to allow components to observe other components' inner states without affecting them. Their state is given by the tuple *<act,0,∅,inst,S,R,F>,* where *S* is the node observed by *R*, and *F* is the function that takes the current state of *S*, stored in *act*, and passes it on to *R*. *inst* is the same as for activation links.

Component links are used to instantiate and remove instances of APOC nodes at runtime (they are the only types of component that can instantiate or terminate an APOC node) and are themselves only instantiated by APOC nodes. Their state is given by the tuple *<act,0,∅,inst,S,R, F>*, where *R* is the node instantiated by *S*, and *F* is the function that takes the instantiation information about *R*, stored in *act*, from *S* and instantiates *R*. *inst* is the same as for activation links.[3]

Properties of APOC

APOC introduces three important conceptual distinctions directly at the architecture level, which are typically absent in agent architectures:[4] a distinction between components (in the architecture layout) and their instances (in the running virtual machine); a distinction between a description of an architecture (or architecture scheme) and a description of an architecture instance (i.e., of a running virtual machine); and, based on that, a distinction between component/ architecture schemes versus components/architectures.

The first distinction seems to be often neglected in agent architectures, presumably because component types as they are depicted in the architecture layout usually have exactly one instance at and throughout the runtime of the virtual machine instantiating the architecture. A planning component in a deliberative layer of a hybrid architecture, for example, which is by definition a type, usually has exactly one instance, which is instantiated once and persists throughout the lifetime of the virtual machine. This neglect, however, deprives agent architectures from being able to describe runtime resources requirements as part of the architecture specification.

The second is a consequence of the first: usually, architectures are instantiated by virtual machines, and hence, there is a type-token relationship between architectures and architecture instances (analogous to the program-process distinction). Some architecture descriptions define architectures, with instances that can be modified over time. These instances might, at some point, cease to be instances of the original architecture *A*, but become instances of an architecture *A**, which can be obtained from *A* using certain operations on the architecture. An example would be growing a neural network, which turns a layered feedforward architecture into an unlayered recurrent architecture.

Finally, the third distinction attempts to tease apart the usage of *architecture* for what could be more appropriately called *architecture scheme* from the usage of *architecture* for a maximally specified control system involving a given set of basic components (namely, the maximal specification of the control system relative to a given level of abstraction, such as the specification of a CPU in terms of logic gates, flip-flops, etc.).

In the following, we will briefly summarize some noteworthy properties of APOC.

Universality

One of the first questions asked about a new formalism concerns its expressive power (compared to standard computational formalisms). Of particular interest is the question of whether the formalism is universal (i.e., can it define a universal machine). Because APOC is a framework, and because its generic components are consequently schematic, this question cannot be directly answered. It is, however, possible to answer it for specific types of components, which can be obtained from generic components by specifying the various parts (update function, sets of input and output states, resource limits, etc.). Depending on how APOC parameters are fixed, the computational power of architecture instances will vary dramatically.

For example, the class of Boolean circuits can be defined by *ACT* = *IN* = *OUT* = *{0,1}*, *PRI* = *{0}*, *PRO* = *{no-proc}*, restricting the update function to mappings from inputs to outputs, and connecting APOC nodes only through activation links with update functions that are the identities on inputs only.

Similarly, all kinds of neural networks can be defined. For example, by changing *ACT* = *IN* = *OUT* = ***Q*** (the set of rational numbers), using a linear threshold summation function as update function restricted to inputs, a universal Turing machine can be implemented, and "Super-Turing" (Copeland, 1998) or *hypercomputations* (Copeland, 2002) can be achieved by using **R** (the set of real numbers) instead of **Q** (Siegelman & Sontag, 1995; Siegelman, 1995). Hence, any computational formalism can be expressed in APOC, and even certain formalisms that transcend Turing-machine computation, such as Turing's *O-machines* (Turing, 1939).

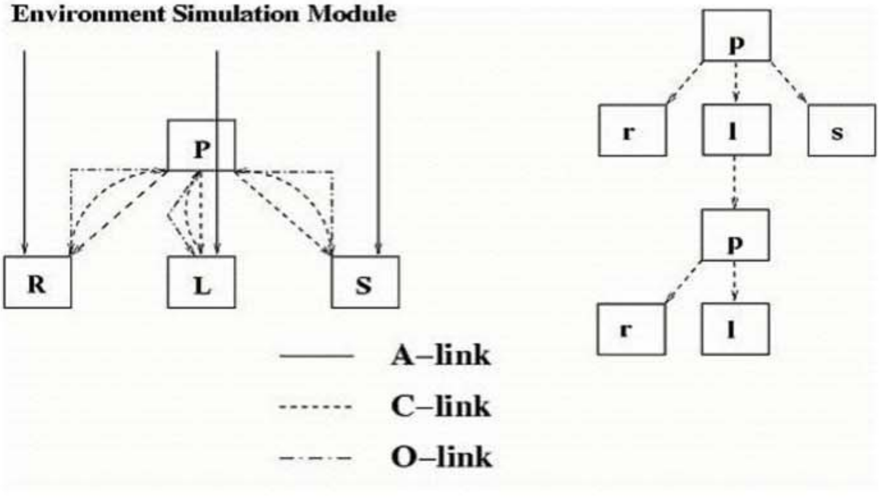
Resource Limitations and Management

APOC allows for explicit resource control and runtime resource management. By virtue of the maximum instantiation number of APOC nodes, the architecture designer can specify ahead of time how many nodes of a given type, at most, can be instantiated in an architecture instance. This can be advantageous, for example, in architectures designed for embodied agents, which have limited memory and processing resources. There, resource utilization is of essential concern (for example, resources that are underused or not used at all should be made available for other purposes).

Another example is resource-dependent planning. Consider an agent that needs to move from its current location to another location by virtue of three actions: *turn right*, *turn left*, and *move straight*. To find the best path, the agent uses a planner, which employs some sort of environment simulation module to simulate the effects of planning steps.

The left part of Figure 1 contains a high-level description of the type relationships in the architecture. The component links in the figure indicate that nodes of type *P* (which encode the planner) can instantiate nodes of types *R*, *L*, and *S* (which represent the respective action types). Conversely, nodes of types *R*, *L*, and *S* can instantiate nodes of type *P*. The implication of this circular component link structure is that nodes of type *P* can instantiate nodes representing actions for all feasible actions up to the resource limit of the system. Note that the action nodes could have processes associated with them that carry out the intended action, thus allowing for a direct mapping from the planning to the plan execution process at the architecture level. The right part in Figure 1 depicts the first three steps of a planning process developed from the structure on the left, where nodes were left uninstantiated in two instances.

Figure 1. Architecture description and instantiation of a typical planner in APOC



Explicit resource control is also helpful in the comparison of different agent architectures as the cost of employing a particular architectural design is brought to light, which might otherwise have been hidden in implementation details of the architectural design.

Translating Architectures into APOC at Different Levels of Abstraction

Given that APOC is universal, any agent architecture can be translated into the APOC framework and consequently analyzed and compared to other architectures in a unified way. However, the translation of an architecture is never unique, but rather multiple translations are always possible, which partly depend on the level at which the original architecture is described and also depend on how primitive components of the architecture are mapped onto APOC components and how connections between those components are expressed in terms of APOC links.[5]

If an architecture is to be described at a low level of abstraction (such as its implementation level), then APOC nodes assume the role of the basic components of the implementing (virtual) machine and do not have associated processes (as in the above example of Boolean networks). This translation usually guarantees the highest degree of parallelism. At higher levels, however, APOC nodes may not be sufficient to specify all details (of the involved processes at lower levels), or the nodes may not be desirable for giving a complete specification of all details. In that case, the associated process of a node can take over the details implicitly, while the node is viewed as and becomes part of a higher-level description. For example, a *behavior* in a behavior-based architecture can be expressed in terms of an APOC node with an associated process that operates on the agent's effectors, while the controlling node reflects the behavior's state and its higher-level activities (for example, participating in action selection).

Another example would be the translation of a cognitive architecture like SOAR, as described in Laird (1987a), which at a high level of abstraction can be described in terms of (a) a working memory (which contains information about *objects*, i.e., goals and states of the system; *preferences*, i.e., structures indicating the acceptability and desirability of objects in a particular circumstance; and a *stack*, which specifies the hierarchy of active goals, problem spaces, states, and operators); (b) a *decision procedure* (which is a function that examines the context and preferences and determines which slot in the context stack requires an action); (c) a *working memory manager* (which determines the elements of the working memory that are irrelevant to the system and deletes them), (d) a *production memory* (which is a set of productions that can examine any part of working memory, add new objects and preferences to it, and add new information to existing objects); and (e) a *chunking mechanism* (which is a learning mechanism for new production rules).

At this level, the APOC translation would roughly consist of one node that implements the working memory and additional nodes that implement the decision procedure, the working memory manager, the production memory, and the chunking mechanism. All nodes would have activation, observer, and priority links to the working memory node. Furthermore, the working memory would have activation links to the production memory, and several additional links would be required to implement the rule-matching and chunking algorithms in SOAR (for example, observer and priority links between the production memory and the chunking mechanism). Note that component links are not required at this level, because all operations that add new facts, production, etc., to the system would occur within an APOC component. Thus, they would not be modeled at the level of the architecture translation, but rather be implicit in the process associated with the respective APOC components.

At a more detailed level, *preferences*, *objects*, and *goals* would be taken to be basic components of an APOC translation, imposing the structure of the SOAR architecture through APOC links. Similarly, the production memory would be mapped at the level of each production as an APOC component. The creation/deletion functionality of SOAR would then map directly onto the APOC component links. However, because the creation process may require additional information to be passed to a newly created node—the conditions in which a new production fires need to be sent to the production when a generic production is created and objects need to be given identifiers—activation links would also need to be created through the component link and be used for information passing. Similarly, the deletion process requires additional information, namely, information about the situation state of working memory (for example, to determine if an object is used in any context on the context stack), which can be retrieved by an additional observer link (that is created by a component link from either the *decision procedure* or the *working memory manager* nodes to a *preference*, *object*, or *goal* node). Even more detailed levels of translation of rule-based systems like SOAR are possible and may be particularly desirable for practical purposes; for more details about translations of common architectures into APOC, see Scheutz and Andronache (in preparation).

Evolving Architectures

In addition to different mappings of components and links of other architectures onto APOC components and links, there is a second way for APOC to allow for abstractions, which has to do with the difference between the architecture specification of the initial state of an architecture and architecture specifications of the modifications of an architecture instance that occur over time. A layered neural network, for example, can be specified by connecting APOC nodes via activation links in two ways: either by instantiating the whole layered network at once, or by defining nodes that will, in turn, construct the layers and then instantiate them. In the latter, the nodes that construct the layers at runtime (by instantiating all nodes that are supposed to be part of the layers together with their connections) can be viewed as representation of these layers.

The left part of Figure 2 depicts the description of a whole class of potential neural network structures: an instance of Type T1 can create 100 instances of Type T2. Similarly, instances of Type T2 can create 100 instances of Type T3. The right part of Figure 2 then shows one possible runtime structure that can be obtained from the type description (depending on the choice of update functions).

This example also demonstrates the bootstrapping process in APOC mentioned earlier, which effectively allows designers to specify growing or evolving structures (such as a layered network that will eventually develop into a fully connected network), in particular, adaptive architectures that can change over time (as a result of some sort of learning process, see Andronache & Scheutz, 2003a). Figures 3 and 4 show an example of such an architecture, an ART network (Carpenter & Grossberg, 1988), which can learn to categorize its inputs over time by adding new categorization nodes to the architecture to represent the newly learned categories. In this example, *E* represents the external inputs, in this case, coming from a sensory node, *G* represents gain control, *T1* represents the node type for nodes in the input layer, *T2* represents the node type for nodes in the category representation layer, and *R* represents the reset of short-term memory.

Figure 2. Architecture description and instantiation of a fully connected layered neural network in APOC

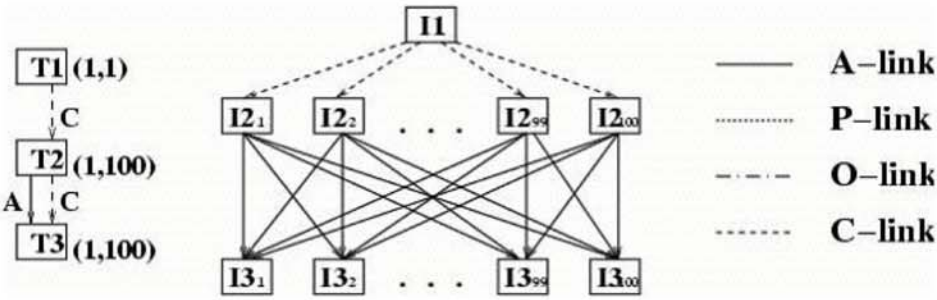
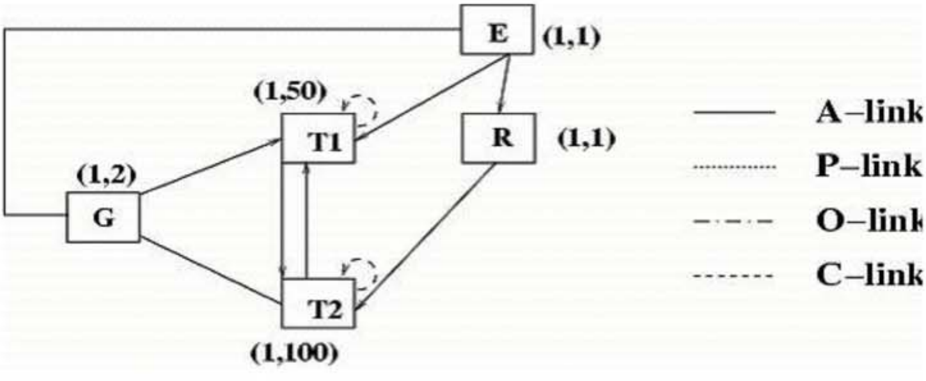


Figure 3. ART type diagram translated into APOC

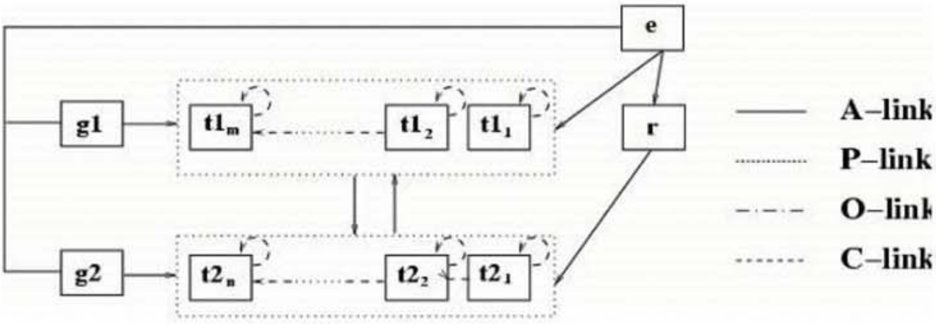


In general, APOC can model dynamic architectures (Andronache & Scheutz, 2003a), where the architecture description changes over time, and architectures in which the basic functional building blocks used to define the architecture are altered as part of the architecture's evolution. This is possible because the update function *F*, which determines the behavior of an APOC component, can be modified so that components and architectures can change. (For example, an APOC component implementing sigmoid update functions of a connectionist unit could change its slope over time, or an APOC component implementing an heuristic planner could change its heuristic over time. In both cases, the input-output mapping of the component is permanently, but not necessarily irreversibly, modified.)

Discussion

While the four APOC link types allow for straightforward translation of many common agent architectures into the APOC framework, APOC is not restricted to software architectures but can also model hardware architectures (basically, any kind of digital design). It is not restricted to architectures for individual agents either but can model multiagent systems as well (at the level of individual perceptions and actions, where each individual agent is modeled by an APOC node, which, in turn, has observer links modeling the perceptions of the agent and activation links modeling the actions of the agent). Moreover, procreation in evolutionary systems can be modeled using component links that allow agents to instantiate copies of themselves (Andronache & Scheutz, 2003a).

Figure 4. Instance of a translated ART type after categorization



Because of its potential to express virtually any kind of agent architecture at several levels of abstraction, APOC seems to be an appropriate formalism for the comparison of different architectures. Architectures can be compared along several dimensions, for example, whether or not they are layered and hierarchical, whether processing proceeds in parallel, whether they are symbolic or nonsymbolic, whether they are for robots or software agents, etc. (see Sloman & Scheutz, 2002, for first steps toward a practical taxonomy of different dimensions of agent architectures).

The mechanisms provided by the four link types as well as the distinction between component description versus component instances may allow for detailed comparisons of different architecture types (for example, a neural network controller versus a subsumption-based finite state machine controller of a robot). Several measures could be defined to aid in such comparisons (one might be the number of APOC components needed to define and implement the basic component of an architecture, the time it takes for a signal to propagate through the network of nodes, etc.). In particular, the proposed notion of cost induced by an architecture can then be used to compare two different architectures for an agent performing the same task. The results of such comparisons can then be used to define new designs that combine advantages of different mechanisms in different circumstances (for example, competitive and cooperative action selection; see Scheutz & Andronache, 2004). It should be noted that the cost of an architecture can also be used to determine the utility of a set of particular components *C* in an architecture *A*, if all other parts of *A* are fixed, and different mechanisms with the same functional and input-output behavior as *C* are substituted for *C* in *A*, yielding the new architecture *A*2, which can then be compared to *A* with respect to cost.

To support the design of agent architectures directly in APOC (without other mediating architectures) and their implementations on simulated and robotic agents, we developed an APOC development environment (ADE) that allows users to develop (i.e., specify, test, and run) specific APOC architectures in JAVA.[6] In ADE, individual APOC nodes can be defined by users targeted at the task at hand and then linked together with a graphical tool using any of the four link types (Andronache & Scheutz, forthcoming). Furthermore, these components can be distributed over multiple hosts to achieve a high degree of real-time parallelism for robotic applications.

Finally, we can return to the venture point about the functional organization of an agent's control system and say something about how functional architectures are related to mechanisms that can implement them. We believe that the way agent architectures can be translated into APOC and defined at different levels of abstraction points in the direction of an answer based on what philosophers call *functional decomposition.* Starting with a high-level functional description of a control system (which typically will have few APOC components with complex associated processes), we can try to recursively relate two functionally equivalent architectural descriptions (of the same control system) at different levels of abstraction by increasingly reducing the complexity of the associated processes of APOC components and making some of their functionality explicit in terms of new APOC components and connections among them that realize this functionality. This successive refinement effectively provides more implementation details about the original (high-level) functional architecture by virtue of functional process decomposition (i.e., by decomposing the associated processes of higher-level components). Eventually, all associated processes will have been eliminated, and the remaining architecture will consist only of APOC components without associated processes. Such an architecture specifies all implementation details necessary to implement it in standard digital and analog circuitry (although it may not be possible to build the hardware for such an architecture, because the architecture is too complex, etc.).

By the same token, the high-level functional description of an APOC architecture can be obtained by successively grouping components to form a new higher-level component, where the functionality internal to the group of components is now performed by the associated process of the higher-level component (the ADE environment provides some support for such grouping operations).

In either case, it is the unified framework that allows for the upward or downward progression through abstraction hierarchies that span the territory from simple low-level (close-to-physical) mechanisms to complex high-level functional architectures. We believe that only by being able to relate different levels of abstraction in a way that is intuitive and meaningful to us as designers of architectures will we be able to understand and build complex agents.

References

Aczel, P. (1988). *Non-well-founded sets. Lecture Notes* (Vol. 14). Stanford, CA: CSLI Publications.

Albus, J. S. (1992). A reference model architecture for intelligent systems design. In P. J. Antsaklis & K. M. Passino (Eds.), *An introduction to intelligent and autonomous control* (pp. 57–64). Boston, MA: Kluwer Academic Publishers.

Anderson, J. R. (1993). *Rules of the mind.* Mahweh, NJ: Erlbaum.

Andronache, V., & Scheutz, M. (2002). Contention scheduling: A viable action-selection mechanism for robotics? In *Proceedings of the Thirteenth Midwest AI and Cognitive Science Conference* (pp. 122–129). Washington, DC: AAAI Press.

Andronache, V., & Scheutz, M. (2003a). Growing agents — An investigation of architectural mechanisms for the specification of "developing" agent architectures. In *Proceedings of FLAIRS2003* (pp. 22–26). Washington, DC: AAAI Press.

Andronache, V., & Scheutz, M. (2003b). *APOC* — A framework for complex agents. In *Proceedings of AAAI Spring Symposium 2003* (pp. 18–25). Washington, DC: AAAI Press.

Andronache, V., & Scheutz, M. (forthcoming). ADE — An architecture development environment for virtual and robotic agents. *International Journal of Artificial Intelligence Tools.*

Arkin, R. C. (1989). Motor schema-based mobile robot navigation. *International Journal of Robotic Research, 8*(4), 92–112.

Barwise, J., & Moss, L. (1996). *Vicious circles.* Stanford, CA: CLSI Publications.

Bonasso, R. P., Firby, R. J., Gat, E., Kortenkamp, D., Miller, D. P., & Slack, M. G. (1997). Experiences with an architecture for intelligent, reactive agents. *Journal of Experimental and Theoretical Artificial Intelligence, 9*(2), 237–256.

Brooks, R. A. (1986). Robust layered control system for a mobile robot. *IEEE Journal of Robotics and Automation, RA-2*(1), 14–23.

Carpenter, G. A., & Grossberg, S. (1988, March). The art of adaptive pattern recognition by a self-organizing neural network. *IEEE Computer*, 77–88.

Chalmers, D. J. (1997). A computational foundation for the study of cognition. (Published on the Internet.)

Copeland, J. (1998). Super Turing-machines. *Complexity, 4*, 30–32.

Copeland, J. (2002). Hypercomputation. *Minds and Machines, 12*, 461–502.

Fikes, R., & Nilson, N. (1971). STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial Intelligence, 2*(3–4), 189–208.

Kim, J. (1996). *Philosophy of mind.* Westview.

Rao, A. S., & Georgeff, M. P. (1991). Modeling agents within a BDI-architecture. In R. Fikes & E. Sandewall (Eds.), *Proceedings of the Second International Conference on Principles of Knowledge Representation and Reasoning* (pp. 473–484). Cambridge, MA: Morgan Kaufmann Publishers.

Scheutz, M. (1999). When physical systems realize functions.... *Minds and Machines, 9*(2), 161–196.

Scheutz, M. (2000). Behavioral states: Linking functional and physical descriptions. In *Proceedings of the AISB'2000 Symposium on How to Design a Functioning Mind* (pp. 117–123). AISB.

Scheutz, M. (2001a). Causal vs. computational complexity? *Minds and Machines, 11*, 534–566.

Scheutz, M. (2001b). Ethology and functionalism: Behavioral descriptions as the link between physical and functional descriptions. *Evolution and Cognition, 7*(2), 164–171.

Scheutz, M., & Andronache, V. (2004). Architectural mechanisms for dynamic changes of behavior selection strategies in behavior-based systems. *IEEE Transactions of System, Man, and Cybernetics Part B: Cybernetics, 34*(6).

Siegelman, H. T. (1995). On computational power of neural networks. *Journal of Computer System Science, 50*(1), 132–150.

Sloman, A. (1998). Damasio, Descartes, alarms and meta-management. In *Proceedings International Conference on Systems, Man, and Cybernetics* (SMC98) (pp. 2652–2657). San Diego, CA: IEEE Press.

Sloman, A. (2000). Architectural requirements for human-like agents both natural and artificial. (What sorts of machines can love?). In K. Dautenhahn (Ed.), *Human cognition and social agent technology, advances in consciousness research* (pp. 163–195). Amsterdam: John Benjamins.

Sloman, A. (2002). Architecture-based conceptions of mind. In P. Gardenfors, K. Kijania-Placek, & J. Wolenski (Eds.), *Proceedings 11th International Congress of Logic, Methodology and Philosophy of Science* [Synthese Library Series] (pp. 397–421). Dordrecht: Kluwer.

Sloman, A., & Scheutz, M. (2002). A framework for comparing agent architectures. In *Proceedings of UKCI'02* (pp. 169–176). U.K.: University of Birmingham.

Turing, A. (1939). Systems of logic based on ordinals. *Proceedings of the London Mathematical Society, 2*(45), 161–228.

Endnotes

[1] The notion of *bisimilarity* is defined as follows: let I and O be two finite sets (for example, the sets of input and output states, respectively), and let $M_1 = \langle S_1, \rightarrow_1 \rangle$ and $M_2 = \langle S_2, \rightarrow_2 \rangle$ be two structures with domains S_1 and S_2, respectively, where relation $\rightarrow_1$ is defined over $S_1 \times I \times S_1 \times O$ and relation $\rightarrow_2$ is defined over $S_2 \times I \times S_2 \times O$. These structures are then said to be *bisimilar* if there exists a nonempty relation R between S_1 to S_2, such that for all $s_1 \in S_1$, $s_2 \in S_2$, $i \in I$, and $o \in O$, the following two conditions hold: if $R(s_1,s_2)$ and $(s_1,i) \rightarrow_1 (t_1,o)$, then $(s_2,i) \rightarrow_2 (t_2,o)$ and $R(t_1,t_2)$; and if $R(s_1,s_2)$ and $(s_2,i) \rightarrow_2 (t_2,o)$, then $(s_1,i) \rightarrow_1 (t_1,o)$ and $R(t_1,t_2)$. For elaboration of the role of bisimulation in a theory of implementation and functional realization (see Scheutz, 2001a).

[2] Links, however, may have multiple input and output ports, and it is permitted in APOC to connect links to links, which proves useful when modeling certain architectures (for example, learning mechanisms in neural networks).

[3] Note that for space reasons, we are not able to provide the details of how the intended component link functionality can be specified in terms of generic APOC components (for example, for each type of APOC node that can be instantiated, a manager node needs to be instantiated to keep track of the total number of instances of that type).

[4] Some architectures incorporate some of these distinctions, such as the distinction between a schema and its instance in schema-based architectures, but not necessarily directly at the architecture level.

[5] Because APOC is not minimal in the sense that any link type can be used to implement all the others (if there are no resource constraints), there are always multiple variations of translation links structures. We view this as a virtue, and not a vice, for one, because it allows designers of agent architectures to use their preferred mechanisms, and second, because it shows how different mechanisms are interrelated.

[6] The ADE tool can be downloaded from www.nd.edu/~airolab/software/

Chapter 15

An Architecture for Cognitive Diversity

Push Singh
MIT Media Laboratory, USA

Marvin Minsky
MIT Media Laboratory, USA

Abstract

To build systems as resourceful and adaptive as people, we must develop cognitive architectures that support great procedural and representational diversity. No single technique is by itself powerful enough to deal with the broad range of domains every ordinary person can understand—even as children, we can effortlessly think about complex problems involving temporal, spatial, physical, bodily, psychological, and social dimensions. In this chapter, we describe a multiagent cognitive architecture that aims for such flexibility. Rather than seeking a best way to organize agents, our architecture supports multiple "ways to think," each a different architectural configuration of agents. Each agent may use a different way to represent and reason with knowledge, and there are special "panalogy" mechanisms that link agents that represent similar ideas in different ways. At the highest level, the architecture is arranged as a matrix of agents: Vertically, the architecture divides into a tower of reflection, including the reactive, *deliberative, reflective, self-reflective, and self-conscious levels; horizontally, the architecture divides along "mental realms," including the temporal, spatial, physical, bodily, social, and psychological realms. Our goal is to build an artificial intelligence (AI) system resourceful enough to combine the advantages of many different ways to think about things, by making use of many types of mechanisms for reasoning, representation, and reflection.*

Introduction

How can we build a machine with the intelligence of a person? There is no shortage of ideas for how to implement in machines aspects of human intelligence, for example, methods for recognizing faces, parsing the syntactic structure of sentences, or planning paths through cluttered spaces. Yet, all such techniques fail miserably in comparison to people when it comes to "commonsense" domains, such as recognizing arbitrary objects in arbitrary scenes, answering questions about the simplest children's story, or stuffing a pillow into a pillowcase. The problem, as we see it, is that the field of AI has focused on solutions to problems that can be captured in the form of single, simple methods, algorithms, and representations, when in fact, the human world is so varied and complicated that any single such solution fails when presented with problems even slightly different from those they were programmed to handle.

How can we build AI systems that are not so fragile? We believe that to build systems as resourceful and adaptive as people, we must develop cognitive architectures that support great procedural and representational diversity. No single technique is by itself powerful enough to deal with the broad range of domains every ordinary person can understand—even as children, we can effortlessly think about complex problems involving temporal, spatial, physical, bodily, psychological, and social dimensions. Ordinary thinking spans so many different types of problems and depends on so many forms of knowledge that unified frameworks, ones that primarily make use of a single type of representation and technique for inferencing and learning, are stretched beyond their capacities. Just as biological systems have no single, simple principle for their operation, we expect that cognitive systems will contain just as numerous and heterogeneous a variety of components.

The *Society of Mind* theory (Minsky, 1986) presents one possible framework for engineering great cognitive diversity. In this theory, the mind is seen as an immense collection of "agents" that perform a wide range of functions, such as expecting, predicting, repairing, remembering, revising, debugging, acting, comparing, generalizing, exemplifying, analogizing, simplifying, and many other cognitive tasks. Agents are not based on any one principle but instead employ a great variety of different methods for learning, representation, and reasoning. In fact, the emphasis in the Society of Mind theory is less on the techniques used by any particular type of agent and more on how groups of these agents can be organized into communities with more capabilities than any individual agent could possibly have. However, the impact of the Society of Mind theory was mixed—while today there is a thriving field concerned with building complex multiagent systems, few such systems aspire to human-level intelligence.

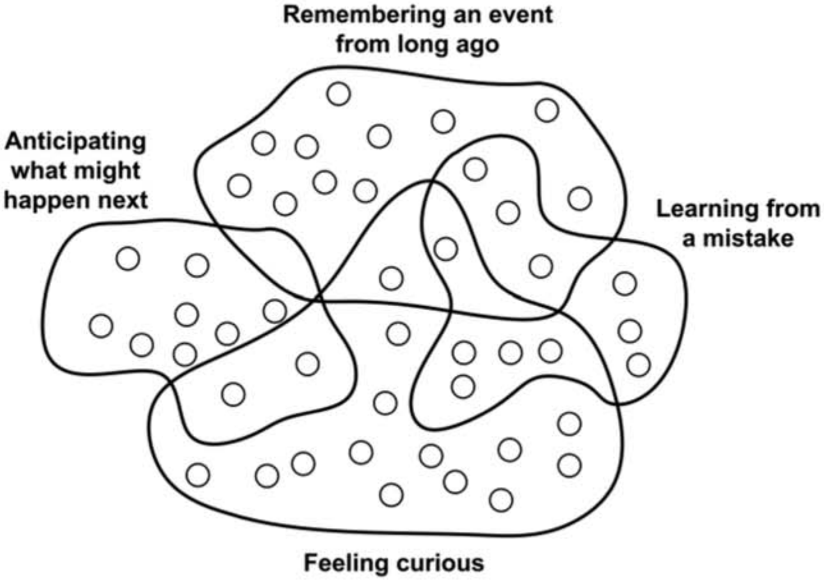
In this chapter, we describe a possible architecture for organizing agents into a flexible, humanlike Society of Mind. Rather than seeking a best way to organize agents, our architecture supports multiple ways to think, each a different architectural configuration of agents. Each agent may use a different way to represent and reason with knowledge, and there are special panalogy mechanisms that link agents that represent similar ideas in different ways. At the highest level, the architecture is arranged as a matrix of agents: Vertically, the architecture divides into a tower of reflection, including the reactive, deliberative, reflective, self-reflective, and self-conscious levels; horizontally, the architecture divides along "mental realms," including the temporal, spatial, physical, bodily, social, and psychological realms. Our goal is to build an AI system resourceful enough to combine the advantages of many different ways to think about things, by making use of many types of mechanisms for reasoning, representation, and reflection.

Ways of Thinking

Our architecture is designed to support a vast diversity of agents, numbering at least in the millions, each roughly on the scale of a small unit of knowledge or subroutine of a computer program. How can we organize a system this large? In our architecture, at any time, only a subset of these agents is active, and each of these states produces a specific way to think. This is illustrated in Figure 1.

In other words, the architecture is not a single kind of "machine," based on a single type of algorithm or method of reasoning. Instead, in different contexts, it transforms into a different machine by switching on different subsets of agents, where the activity of each subset results in a different way of thinking about things. Some examples of these ways to think include the following:

Figure 1. Each way to think results from the activity of a particular subset of mental agents


Remembering an event from long ago
Anticipating what might happen next
Learning from a mistake
Feeling curious

- Solving problems by making analogies to past experiences (for example, Carbonell, 1986)
- Predicting what will happen next by rule-based mental simulations (for example, Kuipers, 1986)
- Constructing new ways to think by building new collections of agents (for example, Minsky, 1980)
- Explaining unexpected events by diagnosis using causal structures (for example, Davis, 1984)
- Learning from problem-solving episodes by debugging semantic networks (for example, Winston, 1970, and Sussman, 1973)
- Classifying types of situations using statistical inference (for example, Pearl, 1988)
- Getting unstuck by reformulating the problem situation (for example, Amarel, 1968)

These ways to think are intended to span the full range of AI methods. At the same time, because each of these ways to think is the result of the activity of a set of agents, new ways to think can be formed by assembling new collections of agents. Ways to think are an evolution of the *K-lines* idea from Minsky's Society of Mind theory (Minsky, 1980). K-lines are special agents with a primary job of switching on other sets of agents. This provides a simple but effective mechanism for disposing a mind toward engaging relevant kinds of problem-solving strategies, retrieving particular fragments of knowledge, selecting or prioritizing sets of goals, invoking memories of particular experiences, and bringing to bear other mental resources that might help in coping with a problem. Each way to think is more or less self-contained, and the mind can be seen as a distributed collection of such ways of thinking with no "central control"—the decentralized vision of cognition presented in the *Society of Mind*.

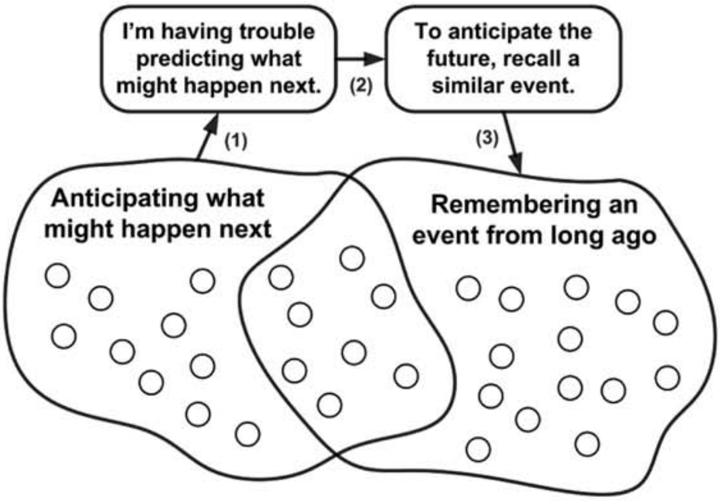
What controls which ways to think are active at any moment? When should the switch be made to new ways to think? In our architecture, there are special "critic" agents concerned primarily with selecting ways to think. At the highest level, these critic agents can be regarded as chronic or persistent questions and concerns, for example:

- What will happen next following this event?
- What would explain why this event occurred?
- What is the best thing for me to do now?
- What can I learn from this failure?
- What might go wrong while performing this action?
- What could be the negative consequences of taking this action?
- Why is that person taking that action?

Each of these mental questions leads to other questions and ways of thinking that can attempt to address them. If we wish to predict what might happen next in a situation, we may try to remember what happened next in a similar situation in the past. If we wish to learn from a failure, we may initiate a credit assignment process that traces back along the causal dependencies among recent events, and so forth.

When a way of thinking begins to fail, the architecture can switch to another more appropriate way to think. This happens through the operation of critic agents that recognize not problems in the outside world, but rather classes of failures and impasses within the mind. When such an impasse is detected, these critics can select alternative ways to think, as shown in Figure 2.

Figure 2. When one way of thinking is beginning to fail, mental critics recognize the failure and select alternative ways to think


I'm having trouble predicting what might happen next.
(2)
To anticipate the future, recall a similar event.
(1)
(3)
Anticipating what might happen next
Remembering an event from long ago

Multiple Representations

When one way to think becomes ineffective, the architecture tries to switch to another. Normally, this would require a certain "startup time," when the agents of the new way to think gather the information they need to do their jobs. However, our architecture performs transitions more efficiently by having special support for multiple representations, to allow agents that represent similar information to easily synchronize what they know. When an agent writes to a representation, it updates the representations of related agents in parallel, including the ones used by agents that are at the moment quiescent. Thus, when the architecture selects a new way to think, instead of having to start from scratch, it finds many of its agents already prepared for the situation.

We do not use any single technique for coordinating representations across multiple agents. Instead, we make use of a family of processes for synchronizing and sharing information. We refer to these together as *panalogy* (a term that derives from "parallel analogy"). Here are some of the methods of panalogy we use:

1. **Event panalogy.** Maintain the correspondences between the elements of action and event descriptions across multiple representations. For example,

when we imagine the consequences of buying a fancy new car, we can rapidly switch between considering the effects of that purchase on our social status (which it may improve) and on our financial situation (which it may hurt.) This form of panalogy lets us assess the consequences of an action or event from a great many different perspectives at once—for in the ordinary, commonsense world, actions and events usually have a wide range of important physical, social, psychological, economic, and other types of consequences.

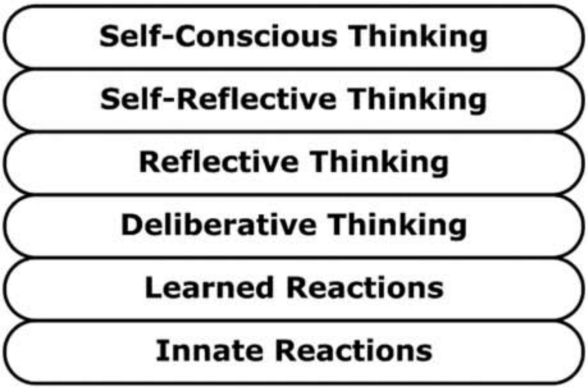
2. **Model panalogy.** Maintain descriptions of different models or interpretations of a situation, like seeing a window simultaneously as an obstacle and as a portal. Each of these interpretations may suggest different inferences or courses of actions, and if we discover that in fact the window is not locked, inferences based on the "portal" interpretation are already available for use. This form of panalogy is valuable, because it takes advantage of the notion that a problem often becomes trivial when we look at it from just the right perspective. A planning problem represented one way might require an immense amount of search, but when represented in another way, it might be solved by simple hill climbing.
3. **Theory panalogy.** Maintain mappings between different theories of the same domain. For example, we may choose to use one theory of time where events are treated as atomic points on a timeline, or use another theory of time where events are treated as occurring over intervals on a timeline. When the first theory is unable to answer a question about, for example, the total duration of some set of actions or the order in which they occurred, we might switch to the second theory. This form of panalogy is useful, because it is difficult to find the best way to represent fundamental commonsense subjects such as space, time, causality, goals, and so forth. We argue instead that there is no best "upper-level ontology" for describing such entities, and that we should instead employ multiple theories about foundational matters.
4. **Realm panalogy.** Maintain analogies between different "mental realms"; large-scale commonsense domains include the spatial, temporal, and social realms. Lakoff and Johnson (1990) have argued, for example, that the knowledge and skills we use for reasoning about space and time are also used to help reason about social realms, for in language, there are pervasive metaphors that exist between these seemingly different domains. This form of panalogy is important, because it is clear from language that it is possible to exploit such metaphors to simplify communication about abstract matters (for example, see the chapter by Barnden, 2004, elsewhere in this volume). We suspect that such metaphors may serve similar roles within the mind as well (see Boroditsky, 2000, for some recent evidence that temporal ideas have their roots in spatial notions).

5. **Abstraction panalogy.** Maintain connections between different abstract descriptions. For example, one might approximate a human skeleton with just a dozen limbs rather than the actual 206 bones of a normal adult, or one might focus on particular subskeletal structures, such as the bones of the right leg. Each of these different abstractions can be linked by their common parts to form a more realistic or complete model than any individual abstraction could form. This form of panalogy is powerful, because it lets us link a variety of "simplifications" of a situation, each useful for a different type of problem. If we are trying to grasp a pair of scissors, it may be useful to think about each of our fingers separately, but if we are trying to push closed a heavy door, we may instead think of the palm of our hand and its five fingers as a single unit that applies pressure to the door.
6. **Ambiguity panalogy.** Maintain links between ambiguous senses of predicates. For example, the preposition "in" can refer to a wide range of relations far more specific than any division provided by ordinary dictionary senses. Rather than select any particular such relation when describing a situation, we can instead maintain the ambiguity between those relations, which then lets us draw on our understanding of all those related senses to answer questions about how one thing could be "in" another. This form of panalogy lets us bypass one of the basic difficulties in building symbolic systems—namely, that it is incredibly challenging and perhaps impossible to define any given symbol precisely enough that we and others will use it only as intended in the future. Just as the meanings of words evolve with their use, and quickly come to acquire multiple new senses in different contexts, so should the meanings of symbols.

Multiple Layers of Reflection

In any multiagent system that regularly faces new situations, the existing community of agents will sooner or later run into problems. An important feature of our architecture is that it is designed to be highly self-reflective and self-aware, so that it can recognize and understand its own capabilities and limitations and debug and improve its abilities over time. In contrast, most architectural designs in recent years have focused mainly on ways to react or deliberate—with no special ability to reflect upon their own behavior or to improve the way they think about things. In our architecture, agents are organized into a tower of reflection consisting of six layers, as shown in Figure 3.

Figure 3. Agents of our architecture are divided into layers, each managing and reflecting upon the layers beneath



Each of these layers is responsible for recognizing and responding to problems within the agents in the layers beneath. To do this, we again make use of special critic agents that bridge these layers. The job of critics is to notice problems in the agents in the layers beneath and select ways to deal with those problems. The function of each layer and examples of the critics that populate each layer are described below:

1. **Innate and learned reactions.** These are instinctive reflexes and learned responses to opportunities and emergencies in the world. They consist of critics that detect specific types of problems in the world and switch on ways to react to those problems. Much of the behavior of animals can be described by networks of such critics. For example:

   I hear a loud noise → Move to a quieter place

   I feel hungry → Follow the smell of food

   I am far from something I want → Walk toward it

   I feel scared → Run quickly to a safe place

2. **Deliberative thinking.** When faced with a difficult problem, it is useful to build a model of the situation in our minds, for example, as a network of goals, actions, and their effects, in which we can search for a solution. The agents of the deliberative layer reason about the situation by engaging in various types of mental deliberation, for example, prediction, explanation, planning, diagnosis, generalization, and so on. Even the simplest problems

may result in large search spaces, and deliberative critics help us search those mazes more effectively:

Action A did not quite achieve my goal → Try harder, or try to find out why

Action A worked but had bad side effects → Try some variant of that action

Achieving Goal X made Goal Y harder → Try them in the opposite order

These events do not chain → Change one of their end points to match

3. **Reflective thinking.** When faced with a tough problem that we are not making much progress on, we may need to reflect on the techniques that we are using to solve that problem. This may involve activities such as assigning credit for success or failure to particular inference methods or types of knowledge, selecting or modifying the knowledge representation structures we have been using, and so forth. Reflective critics assess the performance of recent deliberations in this way and suggest high-level changes to the way we are approaching the current situation:

   The search has become too extensive → Find methods that yield fewer alternatives

   You have tried the same thing several times → Some manager agent is incompetent

   You overlooked some critical feature → Revise the way you described the problem

   You cannot decide which strategy to use → Formulate this as a new problem

4. **Self-reflective thinking.** When reflecting on the methods we use fails to help, we may criticize ourselves. The self-reflective layer is concerned with large-scale models of the "self," including the extent and boundaries of one's physical and cognitive abilities and knowledge. Self-reflective critics look for highly entrenched long-standing deficiencies and weaknesses in our knowledge and methods and suggest significant courses of action to deal with such problems:

   I missed an opportunity by not acting quickly enough → Set up a mental alarm that warns me whenever I am about to do that

   I can never get this exactly right → Spend more time practicing that skill

   I let my other interests take control → Tell one of my friends to scold me when I get distracted

I do not seem to have the knowledge I need → Quit this and go to graduate school

5. **Self-conscious thinking.** It is occasionally useful to imagine what others might think of our activities and how others might approach these same problems. This layer is concerned with the relationship between one's mind and those of others and performs self-appraisals by comparing one's abilities and goals with those of others. Self-conscious critics resemble self-reflective critics, but they operate at a more social level by imagining what others, and especially people whom we respect, might think of us:

   I think I am good at this task → Can I do it as well as the best people I know?

   My mentor would not have made this mistake → What would he have done in this situation?

   Others will think less of me if I keep failing at this → Maybe I should give up doing this sort of thing

   How is it that other people can solve this problem? → Find someone good at this problem and spend time with them

By employing multiple layers of mental critics, we need not build architectures under the impossible constraint that agents always produce the correct inference or perfect suggestion for a course of action. Instead, when the architecture fails, it can examine its own recent activity and self-models to attempt to diagnose and deal with the problem, so that next time, it does not make the same type of mistake.

Many Realms of Commonsense Thought

It is not enough for our architecture to possess many mechanisms for representation, reasoning, and reflection. In addition, it must actually *know* things to cope with the great complexity of the human world. In our view, an architecture that comes with no knowledge is much like a programming language that comes with no libraries or example programs—it is very difficult to get started with it or put it to practical use. What sorts of knowledge should a commonsense architecture possess? If one stops to think about the range of types of things that people know about and the kinds of problems people can solve, it is clear that the list is enormous, and at first glance, it may seem to consist of an entirely haphazard collection of knowledge and skills.

However, while the range of things that an adult human knows about is vast, there is a much more limited class of things that we can expect all people to be able to think about, and especially, the average young child to be able to think about. If we limit the scope of our study in this way, we can approach more systematically the problem of determining what our architecture should know about, how to represent that knowledge, and how to teach it that knowledge. We have been enumerating a list of mental realms, the general commonsense domains that all people, including children, have at least some expertise in. We regard these mental realms as so fundamental that it would be reasonable to regard the inability to reason in terms of one these mental realms as a serious cognitive deficiency.

What are some of the important mental realms? We do not yet have a well-defined, definite list of such realms, but the following are good examples of what we mean by a realm:

- **Spatial.** The spatial realm is concerned with representing the shapes, relative positions, and orientations of places, objects, and their parts. It is also concerned with the motion of objects and the paths that they take through space, as well as the relative spatial relationships between objects as they move. It is the knowledge and processes of the spatial realm we use when solving problems like determining whether objects are close enough to reach, whether it is possible for us to squeeze through a narrow passageway, or how to fit several pieces of wood together to build a table.
- **Physical.** The physical realm is concerned with representing the dynamic behavior of real objects, such as how different objects respond to various forces, perturbations, and other physical interactions. We all know, for example, that you can push things with a stick but cannot pull them with a stick—unless the end of the stick is somehow "attached" to the object we are pulling, or unless the end of the stick is curled into a hook, which lets us convert a pull into a push on the other side.
- **Bodily.** The bodily realm is largely concerned with representing the abilities of your body, such as how far you can reach in different directions from different initial postures, what procedure you should follow to grasp an object of a given shape, or whether you are strong enough to pick up a particularly large object. The bodily realm, combined with the spatial and physical realms, constitutes much of the knowledge a humanoid robot would need to get around in the world and physically manipulate the objects it encounters, such as putting a pillow in a pillowcase, hanging a set of curtains, or throwing a tennis ball to another robot.

- **Social.** The social realm is concerned with representing the relationships, mutual dependencies, and interactions that exist and occur between social entities, such as people and animals. This includes matters such as determining whether your goals are compatible with the goals of others, keeping track of the people you know and the experiences you have shared with them, predicting how someone you know might behave in different situations, and so on. It is the social realm that lets us infer, for example, that someone who is laughing and smiling while talking to someone else is probably enjoying spending time with them.
- **Psychological.** The psychological realm is concerned with representing matters of our own psychology, such as how long it takes us to learn some new subject, whether we are capable of arguing some point or whether we must admit ignorance, the goals that we presently have and their relative priorities, and so on. The psychological realm is about the many types of problems that occur within our own minds. In our view, very little is known about this realm in comparison to many of the other realms we have discussed, simply because to understand this realm well requires a detailed architectural model of the mind.

These are just a few of the realms we have been exploring. We found it useful to further subdivide these realms into more specialized subrealms concerned with more specific matters. For example, in the bodily realm, we may choose to separate knowledge about how to manipulate objects dexterously with our hands from knowledge about how to use our legs to walk over complex terrains.

Each of these realms involves a substantial amount of knowledge. However, realms are distinguished not only by the knowledge they include but also by the methods of reasoning they support. Problems within certain realms may, for efficiency, use specialized representations and reasoning methods. For example, in the spatial realm, a specialized planner that is designed for a three-dimensional Euclidean space may be more suitable to solving spatial path planning problems than some more general technique that can search some arbitrary search space.

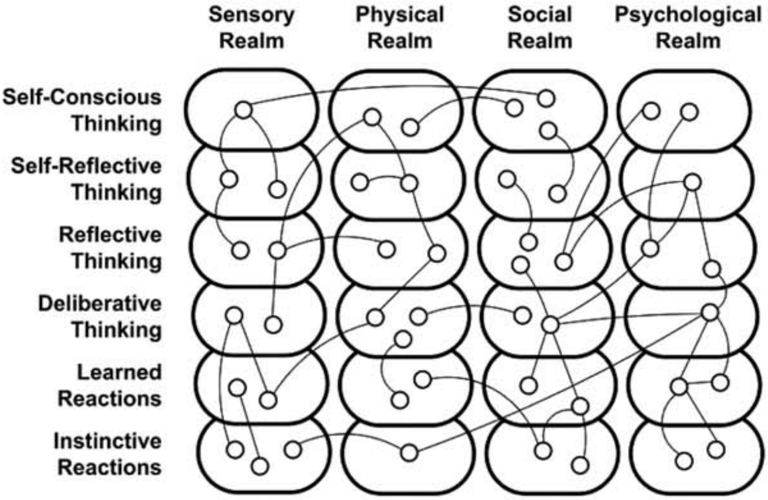
The notion of mental realms has helped to organize the types of commonsense knowledge that our architecture will need. It has also been useful for educational purposes. To the beginner, the idea of commonsense can seem vague and undifferentiated, yet after we introduce to them the concept that commonsense can be divided and organized into large-scale specialties, they often seem to better understand what we mean by commonsense. It should be noted, however, that we have not been using the notion of mental realms in any definite technical sense—that is, a realm does not refer to any particular computational object, except very loosely as that set of representations and processes that are used to cope with a certain wide class of problems.

The Realm-Layer Matrix

We have found it to be useful to merge the previous two ideas of reflective layers and mental realms into the matrix shown in Figure 4. Each cell of the matrix consists of populations of agents that think about a certain mental realm at a certain level of reflection. While there are problems with this diagram—for example, what does it mean for there to be agents in the instinctive-psychological cell?—we have found that more often than not, there seem to be interesting processes at play in each of these cells.

Let us examine a single vertical slice of the architecture, for example, the social realm. At the lowest reactive levels, there are processes for recognizing that someone is smiling at you, for smiling back at them, and so on. At the deliberative level, there may be models of how people react to different sorts of social actions, which includes knowledge, such as someone who smiles kindly at you probably has no malicious intent, or perhaps recognizes you. At the reflective level, there may be processes for understanding why we made a social mistake about classifying our relationship with someone else—for example, it is not always the case that someone who smiles at you knows you, but you had jumped to that conclusion by mistake, and in fact, they were trying to introduce themselves to you. At the self-reflective level, you may decide that you are no good at remembering people's faces and need to do something about that problem. At the

Figure 4: Architecture can be divided into a matrix of cells, for example, the physical-deliberative cell or the social-reflective cell



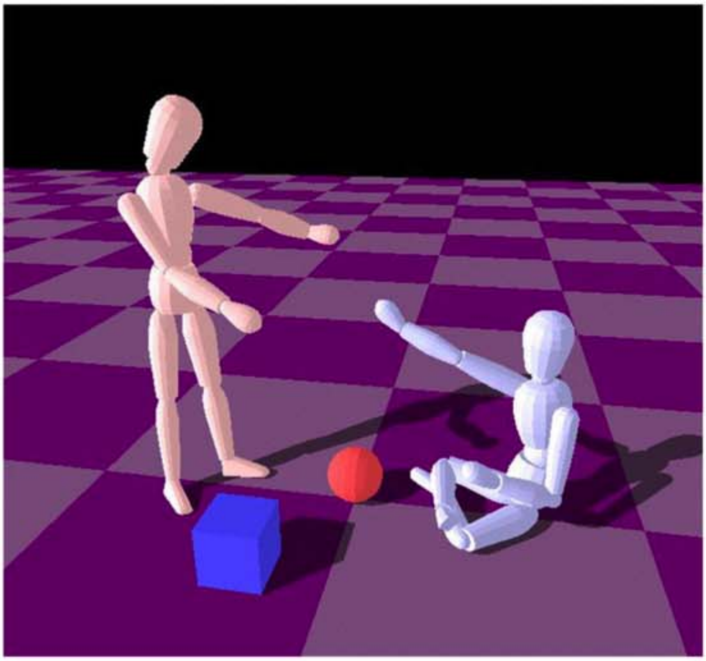
self-conscious level, you may decide that the other person thinks less of you for making that mistake, resulting in a feeling of mild embarrassment.

An Example Scenario

We are developing a concrete implementation of this architecture in the context of an "artificial life" scenario in which two simulated people in a virtual world work together to build complex structures from simple objects like sticks, balls, and blocks, as in the simulator screenshot shown in Figure 5.

While this domain may seem sparse, its simplicity hides a great depth of issues. In particular, the mental realms we discussed so far all show up in some form in this domain. Because the world is physically realistic, the people must reason about the effects of gravity on objects and the forces that must be applied to move them. Because the people have synthetic vision systems, they must reason about whether objects that seem to have disappeared behind bigger ones are still there. Because there are two people, they must reason about the social challenges that arise between them, such as conflicts between their goals and possible opportu-

Figure 5: A simulated world



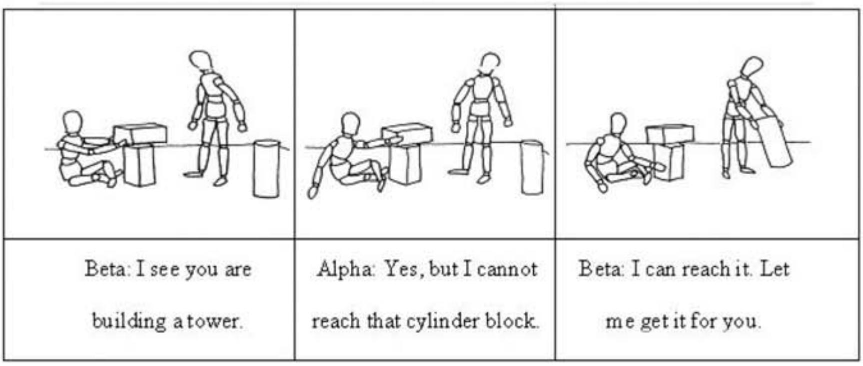
nities for cooperation. To solve problems in this world requires reasoning simultaneously about the physical, social, psychological, and several other mental realms.

Consider the simple scenario shown in Figure 6, depicting two people named Alpha and Beta working together to build a tower. Let us examine Alpha's thoughts during the first two frames of this situation, where it reaches for a block, fails, then realizes that Beta may be able to help. Even this seemingly simple problem requires commonsense reasoning across multiple realms and multiple levels—reactive, deliberative, and reflective processes across the physical, bodily, spatial, perceptual, and social realms:

Frame 1: Alpha wants to build a tower three blocks high. It tries and fails to reach the cylinder block.

- **Deliberative physical (goals):** I want to build a tower of blocks three high.
- **Deliberative physical (situation):** I have built a tower two blocks high.
- **Self-reflective physical (known methods):** What do I know about assembling structures?
- **Deliberative ownership (goals):** I need to obtain a new block to build a tower three high.
- **Reflective priorities (expectation):** If I am building a tower, then I will not be able to achieve my other goals.
- **Deliberative physical (situation):** There are three blocks in this room.
- **Deliberative resources (situation):** Those three blocks are ingredients for the desired tower.
- **Reflective visual and spatial (question):** Perhaps there is another hidden block?
- **Reflective procedural (memory):** I have never built a tower exactly three blocks high.
- **Reflective debugger (method):** Abstracting my goal may result in finding a suitable method.
- **Reflective debugger (method):** Replace "three" with "several."
- **Deliberative spatial (expectation):** Placing a block on top of a tower will make the tower higher.
- **Deliberative bodily (situation):** There is a cylinder block nearby that I can possibly reach.

Figure 6. Alpha (left) builds a tower with Beta (right)


Beta: I see you are building a tower.
Alpha: Yes, but I cannot reach that cylinder block.
Beta: I can reach it. Let me get it for you.

- **Deliberative spatial (expectation):** I might not be able to reach that cylinder block.
- **Reactive bodily (action):** Produce appropriate muscle actuations to produce desired hand trajectory.
- **Deliberative bodily (situation):** My arm is at full length, and I do not have the cylinder block in hand.
- **Reflective bodily (critic):** The current method has completely failed.
- **Self-reflective bodily (method):** Find another method.

Frame 2: Alpha sees Beta and asks for help.

- **Self-reflective bodily (critic):** No method is available for easily reaching that block from here.
- **Self-reflective social (selector):** Switch to social way of thinking for obtaining help.
- **Deliberative spatial (situation):** Beta is nearby.
- **Deliberative social (situation):** Beta may be able to help me.
- **Deliberative spatial (inference):** Beta seems to be near enough to the block to reach it.
- **Deliberative communication (method):** Ask Beta for help.
- **Deliberative psychological (goal):** Beta may have other things to do.

- **Deliberative social (situation):** Beta may want that block for itself.
- **Deliberative linguistic (situation):** Beta says it notices I am building a tower.
- **Deliberative social (inference):** Beta understands my larger goal.
- **Deliberative social (inference):** It would cost Beta very little to help me.
- **Reactive linguistic (action):** Say "I cannot reach that cylinder block."
- **Reactive gestural (action):** Point at cylinder block.

So, we see that even this simple seemingly trivial exchange exercises many of the cells of our matrix of commonsense agents. The details here are greatly simplified—every step involves many more agents than are listed, and many iterations of thought must be involved in producing and refining the solutions to the subproblems encountered by those agents.

Conclusion

This chapter elaborates on a similar discussion in Singh and Minsky (2003). More details about our architectural design are available in Minsky's book *The Emotion Machine* (forthcoming), in McCarthy et al. (2002), and in Minsky et al. (forthcoming).

One might question the need for an architecture with its many lists, catalogs, and other accumulations of kinds of components and features. There will surely be those who find such approaches inelegant, and instead would prefer something simpler, perhaps based on some new mathematical principle or universal method of learning or reasoning. But, our view is that any approach that seeks to build something as complex as a human mind will need to consist of great accumulations of representations and methods, for many of the same reasons that a typical modern computer requires many thousands of small files and programs to operate. Can we realistically expect something comparable to the human mind to be reduced to some simple algorithm or principle given the range of things it must be able do? We hope that our architectural design will change the way AI researchers picture what an AI system should look like and convince people to value systems less on some ethereal notion of elegance and more based on their speed, flexibility, and all-around resourcefulness.

Acknowledgments

We would like to thank Barbara Barry, Ian Eslick, Hugo Liu, Erik Mueller, and Aaron Sloman for many valuable discussions about these ideas. We would like to extend our thanks to the many sponsors of the MIT Media Lab, and in particular, to the J. Epstein Foundation, for supporting our research.

References

Amarel, S. (1968). On representations of problems of reasoning about actions. *Machine Intelligence, 3*(3), 131–171.

Barnden, J. (2004). Metaphor, self-reflection and the nature of mind. In D. N. Davis (Ed.), *Visions of mind*. Hershey, PA: Idea Group.

Boroditsky, L. (2000). Metaphoric structuring: Understanding time through spatial metaphors. *Cognition, 75*(1), 1–28.

Carbonell, J. (1986). Derivational analogy: A theory of reconstructive problem solving and expertise acquisition. In R. Michalski, J. Carbonell, & T. Mitchell (Eds.), *Machine learning: An artificial intelligence approach.* San Mateo, CA: Morgan Kaufman Publishers.

Davis, R. (1984). Diagnostic reasoning based on structure and behavior. *Artificial Intelligence, 24*, 347–410.

Kuipers, B. (1986). Qualitative simulation. *Artificial Intelligence, 29*, 289–338.

Lakoff, G., & Johnson, M. (1990). *Metaphors we live by*. Chicago, IL: University of Chicago Press.

McCarthy, J. (1993). Notes on formalizing context. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI 1993).* Avignon, France.

McCarthy, J., Minsky, M., Sloman, A., Gong, L., Lau, T., Morgenstern, L., Mueller, E., Riecken, D., Singh, M., & Singh, P. (2002). An architecture of diversity for commonsense reasoning. *IBM Systems Journal, 41*(3), 530–539.

Minsky, M. (1980). K-lines, a theory of memory. *Cognitive Science, 4*, 117–133.

Minsky, M. (1986). *The society of mind*. New York: Simon & Schuster.

Minsky, M. (forthcoming). *The emotion machine*.

Minsky, M., Singh, P., & Sloman, A. (forthcoming). The St. Thomas commonsense symposium. *AI Magazine.*

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. San Mateo, CA: Morgan Kaufmann.

Singh, P. (2003). A preliminary collection of reflective critics for layered agent architectures. In *Proceedings of the Safe Agents Workshop (AAMAS 2003*). Melbourne, Australia.

Singh, P., & Minsky, M. (2003). An architecture for combining ways to think. In *Proceedings of the International Conference on Knowledge Intensive Multi-Agent Systems*. Cambridge, MA.

Sussman, G. J. (1973). *A computational model of skill acquisition*. PhD thesis. Department of Mathematics, MIT.

Winston, P. H. (1970). *Learning structural descriptions from examples.* PhD thesis. Department of Electrical Engineering, MIT.

About the Authors

Darryl N. Davis is a lecturer in artificial intelligence at the University of Hull. He graduated from the University of Sussex in experimental psychology, from Heriot-Watt University in knowledge base systems, and from Victoria University of Manchester with a PhD in investigative and diagnostic medicine. He has worked at the University of St. Andrews on human visual perception, particularly human face recognition. At the University of Manchester, he worked on artificial intelligence architectures for medical image interpretation. At the University of Birmingham, he was employed on a number of projects, including one within the Cognition and Affect group. He has been a JSPS funded visiting professor at Kyoto University. His current research focus is on architectures for cognition and affect, adaptive segmentation in machine vision, and data mining in medicine.

* * *

Andy Adamatzky is a reader in the Faculty of Computing, Engineering and Mathematical Sciences, University of the West of England, Bristol. He does research in unconventional computing, cellular automata, massive parallel computing, applied mathematics, theoretical computer science, collective intelligence, and robotics. He authored *Identification of Cellular Automata* (Taylor & Francis, 1994) and *Computing in Nonlinear Media and Automata Collectives* (IoP Publishing, 2001), compiled *Collision-Based Computing* (Springer, 2002), and coedited *Molecular Computing* (MIT Press, 2003).

Michel Aubé is professor at the Université de Sherbrooke, Canada, since 1983, where he teaches science education, teacher training, and the integration of computers in learning. He obtained his master's degree in cognitive psychology from the University of Toronto, and his PhD in education sciences from the Université de Montréal. His doctoral studies bear upon the design of a computational model of emotions, called "The Commitment Theory of Emotions." He is also involved in the design and implementation of an educational Web site, "Le Monde de Darwin," dedicated to the scientific training of children from primary schools, by putting them in contact with adult scientists. His research interests are distributed along three main axes: use of computer technologies to foster scientific thinking in children; development of a robust computational model of emotions; and use of computer technologies in building online distance-training systems for teachers.

John A. Barnden has been a professor of artificial intelligence at the University of Birmingham since 1997 and head of the School of Computer Science there since 2000. His previous position was at the Computer Science Department and Computing Research Laboratory, New Mexico State University (1987-1997). He has a first degree in mathematics from Cambridge University (UK) and a doctorate in artificial intelligence from Oxford University (UK). His research interests include metaphor understanding, communication and reasoning about mental states, and the application of connectionism to reasoning. He is currently chair of the Society for the Study of Artificial Intelligence and the Simulation of Behaviour.

Joanna J. Bryson has been exploring the interaction between modularity and intelligence since 1992. She holds a PhD from MIT in computer science, a master's degree in psychology and artificial intelligence from Edinburgh, and a BA in behavioural science from Chicago. In 2002, she founded the artificial models of natural intelligence (AmonI) group at the University of Bath. The group's goal is to popularise the use of artificial intelligence as a tool for scientific research, both by example and by software platform development.

After graduating from Evergreen College (USA) in 1976, **William F. Clocksin** did four years of postgraduate research in computer vision, robotics, and logic programming at the Department of Artificial Intelligence, University of Edinburgh. From 1980-1983, he worked in a robotics research group at the Engineering Science Department, University of Oxford, and was a fellow of St. Cross College, Oxford. From 1983-2001, he was a university lecturer at the Computer Laboratory, University of Cambridge, where he carried out teaching and research in artificial intelligence. From 1987-1999, he was a fellow of Trinity

Hall, Cambridge, where he also served as assistant chaplain from 1993-2001, and acting dean from July 2000 to September 2001. He was appointed professor of computer science at the Department of Computing at Oxford Brookes University in October 2001. His research interests include the application of computer vision and image processing to real-world problems in medicine and factory automation, and the development of a social constructionist framework for artificial intelligence.

Bruce Edmonds is a senior research fellow at the Centre for Policy Modelling at the Manchester Metropolitan University. His first degree was in mathematics, and his PhD was in the philosophy of science. Although he is interested in (too) many areas — his research is centred on agent-based social simulation, which lies at the crossroads of artificial intelligence/software agents and cognitive science/sociology. That is, techniques from computer science are used to simulate social phenomena, and ideas from social phenomena are used in the construction of new computational systems. A complete list of his publications (many of which are online) can be found at *http://cfpm.org/~bruce*.

John Fox was educated at Durham and Cambridge Universities, UK, and then carried out research in artificial intelligence and cognitive science at Carnegie-Mellon and Cornell Universities (USA). On returning to the UK, he took up a post with the Medical Research Council, for which he worked on theory and technologies for clinical decision making. In 1981, he joined the Imperial Cancer Research Fund's Laboratories in London (now Cancer Research UK), where he set up the Advanced Computation Laboratory. The laboratory's research straddles computer science, artificial intelligence and cognitive science, and medical software engineering. Fox has published widely in these fields and was founding editor of the *Knowledge Engineering Review* (Cambridge University Press). Recent publications include *Safe and Sound: Artificial Intelligence in Hazardous Applications* (AAAI and MIT Press, 2000), which deals with many aspects of the use of AI in medicine and other safety-critical fields and underlying theoretical issues in cognitive science.

A mathematician turned computer scientist turning cognitive scientist, **Stan Franklin** is Dunavant professor of computer science at the University of Memphis and is codirector of its Institute for Intelligent Systems. His research is motivated by wanting to know how minds work — human minds, animal minds, and, particularly, artificial minds. For some years, he has worked on "conscious" software agents, that is, autonomous agents modelling a psychological theory of consciousness. His graduate degrees are from UCLA, and his undergraduate degree is from the University of Memphis. He has authored or coauthored some 80 papers and one book entitled *Artificial Minds*.

David W. Glasspool carries out research in planning and decision making at the Advanced Computation Laboratory of Cancer Research UK, where he has worked since 1998. He holds a PhD in computational modelling from the Psychology Department of University College London and an MSc in cognitive science from Manchester. His main research interests are in executive control, routine behaviour, and planning in the face of uncertainty, both in computational systems and in human cognitive psychology.

Fernand Gobet earned his PhD in psychology at the University of Fribourg, Switzerland. After a stay at Carnegie-Mellon in Pittsburgh (USA), as a visiting researcher and at the University of Nottingham, UK, as a reader in intelligent systems, he is currently professor of psychology at Brunel University. He has written four books on expertise and computational modelling. In his current research, he uses computer modelling and brain imaging to study experts' cognitive processes, human learning and memory, and the acquisition of language. He is also interested in the foundations of computational modelling, including the use of artificial intelligence techniques for semiautomatically developing scientific theories.

Elizabeth Gordon is currently completing a PhD at the University of Nottingham. Her work is sponsored by Sony Computer Entertainment Europe. She holds a Bachelor of Science in computer science from Harvey Mudd College, California, and a Master of Science in artificial intelligence from the University of Edinburgh.

Pentti O. A. Haikonen received an MSc (EE), Lic in Tech and DrTech from the Helsinki University of Technology, Finland (1972, 1982, and 1999, respectively). He is currently the principal scientist in the area of cognitive technology in Nokia Research Center, Helsinki, Finland. He is the author of the book *The Cognitive Approach to Conscious Machines* (UK, Imprint Academic, 2003). He has several patents on video signal processing, associative neurons, and networks. His research and hobby interests include machine cognition, electronic circuitry for cognition, and the design of exotic electronic circuits.

Colin G. Johnson is lecturer in computer science at the University of Kent. Prior to this, he worked at the University of Exeter and Napier University in

Edinburgh and was a student of mathematics at the University of York. His research interests are centred on the relationship between computing and the natural sciences. On one hand, this work focuses on applying computational concepts to problems in the biological, medical, and cognitive sciences, for example, through the development of simulations of biological systems and in using analogies from computer science to understand the functioning of complex systems. The other aspect of this work is concerned with drawing inspiration from natural science to develop novel ways of doing computation.

Peter C. R. Lane is a lecturer in computer science at the University of Hertfordshire, UK. He received a BA in mathematics and computation from the University of Oxford (1991), an MSc in computer science from the University of Exeter (1995), and a PhD in computer science from the University of Exeter (2000). His research interests are centred on topics in machine and human learning, including the CHREST model of human perception and learning, computational linguistics, connectionist networks, and machine vision.

Brian Logan is a lecturer in the School of Computer Science and IT at the University of Nottingham, UK. He received a PhD in design theory from the University of Strathclyde, UK (1986). His research interests lie in the area of agent-based systems, and span the specification, design, and implementation of agents, including agent architectures, logics, and ontologies for agent-based systems and software tools for building agents. Before moving to Nottingham, he was a member of the Cognition and Affect Group at the University of Birmingham, where he worked on architectures for autonomous intelligent agents capable of complex decision making under constraints such as incomplete and uncertain information and time pressures.

Marvin Minsky has made many contributions to artificial intelligence, cognitive psychology, mathematics, computational linguistics, robotics, and optics. In recent years, he has worked chiefly on imparting to machines the human capacity for commonsense reasoning. His conception of human intellectual structure and function is presented in *The Society of Mind*, which is also the title of the course he teaches at MIT. He received a BA and PhD in mathematics at Harvard and Princeton. In 1951, he built the SNARC, the first neural network simulator. His other inventions include mechanical hands and other robotic devices, the confocal scanning microscope, the "Muse" synthesizer for musical variations (with E. Fredkin), and the first LOGO "turtle" (with S. Papert). A member of the NAS, NAE, and Argentine NAS, he has received the ACM Turing Award, the MIT Killian Award, the Japan Prize, the IJCAI Research Excellence Award, the

Rank Prize and the Robert Wood Prize for Optoelectronics, and the Benjamin Franklin Medal.

Matthias Scheutz received a PhD in philosophy from the University of Vienna (1995) and a joint PhD in cognitive science and computer science from Indiana University, Bloomington (1999). He was a postdoctoral research fellow in the Cognition and Affect Group at the University of Birmingham, UK, working with Aaron Sloman on affect and cognitive architectures. He started his current position as an assistant professor at the University of Notre Dame in 2001, where he now leads the Artificial Intelligence and Robotics Laboratory in the Department of Computer Science (*http://www.nd.edu/~airolab/*). He has published more than 50 peer-reviewed articles on topics including the utility of affective mechanisms, architecture frameworks for agent architectures (in particular, complex cognitive agents), and foundations of cognitive science. His current interests focus on a new general architecture framework called APOC, in which all agent architectures can be defined and studied in a unified way.

Push Singh is a doctoral candidate in MIT's Department of Electrical Engineering and Computer Science. His research is focused on finding ways to give computers human-like commonsense, and he is presently collaborating with Marvin Minsky to develop an architecture for commonsense thinking that makes use of many types of mechanisms for reasoning, representation, and reflection. He started the Open Mind Common Sense Project at MIT, an effort to build large-scale commonsense knowledge bases by turning to the general public, and has worked on incorporating commonsense reasoning into a variety of real-world applications. Singh received his BS and MEng degrees in electrical engineering and computer science from MIT.

Index