

**SITUATED REPRESENTATION:
SOLVING THE HANDCODING PROBLEM
WITH EMERGENT STRUCTURED REPRESENTATION**

BY

CLAYTON THOMAS MORRISON

BA, Occidental College, 1992
MA, Binghamton University, 1995

DISSERTATION

Submitted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Philosophy
in the Graduate School of
Binghamton University
State University of New York
1998

© Copyright by Clayton Thomas Morrison 1998
All Rights Reserved.

The Far Side cartoon by Gary Larson appearing on Page 1:
THE FAR SIDE ©1987 FARWORKS, INC.
Used by Permission of UNIVERSAL PRESS SYNDICATE.
All Rights Reserved.

Accepted in partial fulfillment of the requirements for
the degree of Doctor of Philosophy in Philosophy
in the Graduate School of
Binghamton University
State University of New York
1998

Eric Dietrich, Chair _____ July 8, 1998
Department of Philosophy
Binghamton University

Mark H. Bickhard _____ July 8, 1998
Department of Philosophy, Psychology and Cognitive Science
Lehigh University

Rom Harré _____ July 8, 1998
Department of Philosophy, Binghamton University
Department of Psychology, Georgetown University

Eileen C. Way _____ July 8, 1998
Department of Philosophy
Binghamton University

Abstract

Cognitive science and artificial intelligence currently lack a robust account of the emergence and change of structured representation. This is a result of limiting assumptions about the nature of representation: what makes a representation *about* something else. These limiting assumptions are reflected in methodological approaches to the modeling of cognitive agents that require any representational components of the agent to be placed in the model — *hardcoded* — by the creator of the model. This handcoding precludes the possibility of an explanation of representation emergence and change. I provide an outline of how to characterize handcoding and argue for why we must take the issue of handcoding seriously or risk losing the explanatory power of our models.

I use the Structure-Mapping Theory (Gentner, 1983, 1989) and High-level Perception (Chalmers *et al.*, 1992; Hofstadter, 1995) models of analogical cognition as a case study to demonstrate the successes and utility of structured representation-based explanation, and to highlight current limits with respect to accounting for fundamental representation emergence and change. I demonstrate how each model relies upon an antecedently fixed *representational grammar* that has to be hardcoded by the creator of the model. This grammar constitutes the fundamental representational building-blocks available for any representation construction or manipulation in the model. These stable, content-identity-bearing atomic units cannot themselves emerge and change. I present evidence, however, that strongly suggests that such emergence and change must be possible, and is directly implicated in analogy-making. The current dependence on handcoding of representational grammars therefore precludes the possibility of these models accounting for the central role of fundamental representation emergence and change in analogical cognition.

I utilize Mark H. Bickhard's (Bickhard, 1993; Bickhard & Terveen, 1995) insights into the nature of representation and his *interactive* model to escape the non-emergence impasse faced by current models. I develop and augment his model to propose an initial account of the possibility for the emergence and change of structured representation in an artificial life model. The result is the *situated representation* framework: a proposal for how structured representational content can emerge and develop in autonomous agents.

Dedication

For Heather and our family

*“When we try to pick out
anything by itself, we find it
hitched to everything else
in the universe”*

– John Muir, 1869

Acknowledgments

The completion of this project would not have been possible without the guidance and support of many people. I am especially grateful to Eric Dietrich for his instruction, vision and counseling. He has been both a mentor and a friend. I am also very indebted to Mark Bickhard; his work has been a continuous inspiration, and his gentle guidance during this project has been invaluable. Eileen Way has also been an invaluable source of insight, ideas and direction; throughout this project she has helped me to uncover crucial concepts that lay implicit in what I have been trying to express. It has been a distinct honor to have had the opportunity to study with and have guidance from Rom Harré, whose comments and direction have always been illuminating. Deep thanks is also due to H. Stephen Straight for providing new perspective and a most thorough proof reading! (Any lingering typos are solely my responsibility!) I owe a great deal of thanks to many others who have labored through my disjoint ideas; our discussions (sometimes heated — but all the better!) have been immeasurably helpful and I hope that I will always have access to the kind of high caliber intellectual community that I have been privileged to participate in over the course of my graduate studies. I would especially like to thank Art Markman, Jerry Aronson, Larry Roberts, Changsin Lee, Doug Beyer, Bran van Heulveln, and Lewis Loren. And I also want to give special thanks to Bobbi Libous for her constant friendship and encouragement.

None of this work would have been possible without the love, support and encouragement of my family. Most important of all has been my wife, Heather, to whom this work is dedicated. She has been the greatest source of inspiration, encouragement and love. Also very important to me has been my new family: Amy, Matt, Myles (in three years' time he's managed to accomplish what none of our computers are likely to do for many years!), Stacey, and my new mom, Kathy. And, of course, my parents, Meg and Steve, and my sister, Katie, have been a crucial and consistent source of encouragement, love and support. I am grateful for the sacrifices they have made so that I could have the opportunity to carry out my studies.

Table of Contents

Abstract	iv
Dedication	v
Acknowledgments	vi
Table of Contents	vii
List of Tables	xii
List of Figures	xiii
Introduction	2
The situated representation project:	
Illustrating the central problem and its solution.....	4
<i>Representation and analogy</i>	6
<i>The problem: handcoding representational content emergence and change</i>	7
<i>Avoiding handcoding: situated cognition and interactive representation</i>	8
The dissertation chapters	12
<i>A diagrammatic tour</i>	12
Chapter 1 - The Handcoding Problem	14
1 - Introduction: The Challenge of Computational Creativity.....	14
2 - A first-pass definition and an intuitive example of “Handcoding”	16
2.1 - An Intuitive Example of Handcoding.....	17
3 - Handcoding in general.....	24
3.1 - The Analog of Handcoding in Other Sciences.....	26
3.1.1 - Direct Researcher Influence	26
3.1.2 - Interpretation	27
3.1.3 - Why is Handcoding So Non-Obvious for Cognitive Science?	31

4 - Three Handcoding Case-Studies	33
4.1 - AM & EURISKO	33
4.2 - BACON.....	36
4.3 - A Connectionist Past-tense Learner: Rumelhart & McClellon's past-tense learning network	39
5 - Summary.....	41
Chapter 2 - Handcoding and Computational Cognitive Modeling	43
1 - Introduction.....	43
2 - Scientific Models	44
2.1 - Some preliminary distinctions.....	45
2.2 - Models	46
2.3 - Measuring Theories	49
2.4 - The relation between models, theories and the world	54
(1) The model's intended use: explanatory task and goal.....	55
(2) The relationship of the model to the world: similarities and differences	55
(3) The semantics of the model	57
(4) Accuracy	58
3 - Computationalism	60
3.1 - The computational hypothesis.....	60
3.2 - Turing-computation and the Church-Turing Thesis.....	62
3.3 - The Theses of Computational Sufficiency and Explanation.....	65
3.4 - Theory of Interpretation.....	66
3.5 - Supervening Machines: computational descriptions become models.....	71
4 - Computationalism's minimal ontological commitments to cognition	77
5 - Framework for the handcoding critique as a methodological tool in computational cognitive modeling.....	81

Chapter 3 - Handcoding and Representation:	
Analogical Cognition as a Case Study	88
1 - Introduction.....	88
1.1 - Initializing the handcoding framework.....	90
1.2 - The outline and tasks of Chapter 3	91
2 - Analogy & Representation	92
3 - Similarity and analogical cognition	95
4 - Structure-Mapping Theory	100
4.1 - SMT's Representational Assumptions	101
4.2 - The Process of Structural Alignment and Mapping	103
4.3 - The family of Structure-Mapping Engine models.....	107
4.3.1 - SME, the core model of the family.....	108
4.3.2 - SME, part of a larger architecture.....	112
4.3.3 - MAC/FAC.....	113
4.3.4 - I-SME.....	116
5 - Conceptual Fluidity and Analogy as High-Level Perception.....	117
5.1 - High-level perception and flexibility.....	118
<i>Low-level vs. High-level perception</i>	118
<i>Flexibility</i>	119
5.2 - The motivation for HLP: avoid handcoding of representation	120
<i>Some problems with the Chalmers et al.</i> <i>approach to handcoding, and SMT's rebuttals</i>	125
5.3 - The architecture for conceptual fluidity and HLP.....	128
<i>Slippage and the relational structure of concepts</i>	128
<i>Codelets and the Parallel Terraced Scan</i>	130
<i>The Slipnet</i>	133
<i>Putting it all together</i>	135
5.4 - The HLP theory of representation.....	137

6 - Handcoding of representation in the SMT and HLP models.....	142
6.1 - “Content” ... and determining it in computational models	142
6.2 - Wishful Mnemonics and The Eliza effect	143
6.3 - Handcoding in Class 1: SME’s representations	147
6.4 - Handcoding in Class 2: high-level perception approaches to representation	152
<i>Treatments of representational grammars and domain-style modeling</i>	154
<i>Handcoding</i>	156
<i>Handcoding of the kind Chalmers et al. identify in SME</i>	157
<i>Handcoding with respect to the emergence or change of the representation grammar</i>	158
<i>What emergence or change might be in the HLP approach to representation</i>	158
<i>A final comment on the treatment of handcoding and learning by Chalmers et al.</i>	159
7 - The Relational Shift in development, Knowledge Change, and Phineas.....	160
7.1 - The relational shift.....	160
7.2 - The SME simulations of the relational shift	164
7.3 - SME’s limits - The <i>What</i> and the <i>When</i> , but not the <i>How</i> of SME.....	165
<i>Gaps in the explanation of representation development</i>	165
<i>SME-based approaches to representation construction and re-representation</i>	167
<i>Can HLP account for the blocked-experiment relational shift?</i>	170
7.4 - Phineas and the skolem predicate/process	170
8 - Summary of the shared representational assumptions and handcoding.....	176
Chapter 4 - The Foundations for Representation Emergence & Change	179
1 - Introduction.....	179
<i>Layout of the Chapter</i>	180
2 - Emergence	182
3 - What representation is for — features of cognitive representation	187

4 - Two approaches to the nature of representation	188
4.1 - Encodingism - the fundamental assumption and its entailments.....	189
<i>The assumption of encodingism</i>	189
<i>The problem with encodingism</i>	191
<i>The results of encodingism's incoherence</i>	192
<i>Encodingism, handcoding and the representational grammar</i>	193
4.2 - Interactivism.....	196
5 - Situated Representation	206
5.1 - The development of level 1: base indicator assemblies and chains	207
5.2 - The development of level 2: chain differentiators.....	212
5.3 - Name bindings and finding invariances: level 3 and beyond.	214
6 - Conclusion	221
<i>What about representation of relations?</i>	221
<i>Representation of structural properties of the world</i>	222
<i>Levels of situated representation:</i>	
<i>recursive complexity vs. knowing levels</i>	223
<i>Have I completely avoided handcoding representation emergence and change?</i>	226
<i>Summary of key results</i>	228
Bibliography	231
Short Vita	248

List of Tables

Table 1 - Different kinds of similarity	96
Table 2 - Ascending Levels of Representational Power	220

List of Figures

(Illustration) The Far Side	1
Figure I.1 - Diagrammatic tour of the dissertation territory.....	13
Figure 1.1 - Representation of the Problem	19
Figure 1.2 - The Loop	20
Figure 1.3 - Repetitions Through The Loop.....	21
Figure 1.4 - The Answer	21
Figure 1.5 - Two Kinds of Handcoding	26
Figure 1.6 - The “Duck-Rabbit” and Necker Cube	28
Figure 2.1 - Giere’s Analysis of Models.....	52
Figure 2.2 - The Role of the Theory of Interpretation	54
Figure 2.3 - Mapping Physical To Machine State-Changes.....	70
Figure 2.4 - Analysis of SVM model of “naturally occurring SVM”	84
Figure 2.5 - Analysis of handcoding of SVM model.....	85
Figure 2.6 - Analysis of handcoding with respect to interpretation	86
Figure 3.1 - Analogy as creative and structured.....	94
Figure 3.2 - Similarity space	96
Figure 3.3 - Mapping M of <i>Source</i> to <i>Target</i>	103
Figure 3.4 - Mapping Rules.....	104
Figure 3.5 - Example of physical situations involving (a) water flow and (b) heat flow.....	109
Figure 3.6 - Representation of water and heat given to SME	110
Figure 3.7 - Postulation of increased structure in target representation.....	111

Figure 3.8 - An architecture for analogical reasoning	112
Figure 3.9 - Ambiguous ‘A’ character in different contexts.....	120
Figure 3.10 - The High-level Perception Architecture	137
Figure 3.11 - Partial Analogy-Square	137
Figure 3.12 - Representation structures in the Workspace of Copycat	140
Figure 3.13 - Changing labels in an SME-style representation.....	147
Figure 3.14 - Different revolves-around situations.....	150
Figure 3.15 - Class 2 models	152
Figure 3.16 - “abc” example.....	154
Figure 3.17 - What the altered Copycat ‘sees’ of the “abc” example.....	154
Figure 3.18 - Example of trial presentation.....	163
Figure 3.19 - Example of Same-dimension vs. Cross-dimension trials.....	163
Figure 3.20 - SME’s time-slice view of representation-development	166
Figure 4.1 - The Cycle of <i>Interaction</i> — the <i>functional interactive loop</i>	201
Figure 4.2 - Representational Content as <i>Indications of Potential Further Interactions</i>	202
Figure 4.3 - An Indicator Assembly for Successful Interaction with Object1	212

Introduction

We have reached an interesting vantage point in the development of cognitive science: we are now in a position to step back and take stock of what can confidently be considered as preliminary success stories in computational explanations of cognitive phenomena; at the same time, we also have enough experience, insight, and history of intellectual ferment to uncover current obstacles and the new directions that research needs to take for future success and growth of the field. The impetus for this dissertation is born directly out of this perspective. In particular, this dissertation is aimed at developing a framework for exposing and escaping an existing problem in computational accounts of representation, cognitive science's primary explanatory construct.

A tension is produced by the juxtaposition of two very promising (possibly even necessary), yet currently non-compatible directions of research. On the one hand, our best explanations of higher-level cognitive capacity require representations with structural complexity. For example, in research on the ability to produce and understand analogies, a large amount of predictive and explanatory success has been garnered through the use of models which assume the existence of structured representations of knowledge. This general approach is referred to as the *symbolic representation* approach because of its reliance on discrete, meaningful symbolic atoms which are related to one-another via labeled (and therefore, also meaning-bearing) links.

Powerful critiques, however, have been leveled against the symbolic representation approach. Many of these initially stemmed from connectionist and parallel distributed processing (PDP) approaches to computation, which offered an alternative perspective to modeling representation (e.g., "subsymbolic" semantics and distributed representations; Smolensky, 1988; Clark, 1989, 1993). More recently, however, critiques have been made on the grounds that the representation schemes assumed are simply not plausible when considered in the context of every-day coping and interaction with the world; these critiques particularly focus on the computational entailments which follow from the assumption of the unproblematic presence of the meaningful, stable content they are intended to "carry" (Agre & Chapman, 1987; Brooks, 1991; Suchman, 1987). Some have even argued that certain assumptions of the symbolic representation approach (*and* approaches to representation in connectionism) are not just practically untenable, but have *in principle* problems which cannot be solved within current symbolic approaches (Bickhard, 1993; Bickhard & Terveen, 1995; Hendriks-Jansen, 1996).

Alternative models have been proposed which have had success at demonstrating that a surprising amount of complex structured behavior can arise out of simple, interactive autonomous agents situated in an environment (e.g., Beer, 1990; Brooks, 1989; Mataric, 1992; and Mataric & Brooks, 1990). This approach is referred to as the *situated cognition* approach (Clancey, 1997; Hendriks-Jansen, 1996). The rub is that situated

cognition currently offers no cohesive account of representation, let alone representations with complex structure. Coupled with the fact that current situated cognition models can only account for relatively low-level behavior compared to the cognitive capacity of, for example, analogical cognition, this leaves good reason to be sceptical that this approach will be able to offer insightful explanations of higher-level cognition.

I propose that the present tension is indicative of an obstacle common to both the symbolic representation approach and situated cognition: *a misconception of the fundamental nature of representation*. What we need is a radical overhaul of our classical conception of representation with structure to produce a new representational framework which will accord with situated cognition's perspectives while preserving our well-supported insights from cognitive psychological research on high-level cognition. In this new framework, the modeling of autonomous agents that must cope with their environment will ground the relation between higher-level cognition and the world, thus avoiding the criticisms that the symbolic representation approach faces; at the same time, these representations will also give the situated cognition approach its needed explanatory power for accounts of phenomena which currently require structured representations.

Unfortunately, a fully developed framework such as this is still a distant goal. Nonetheless, several of the key foundational pieces for this overhaul have already been developed. In particular, Mark H. Bickhard, in his work on the fundamental nature of representation (Bickhard, 1993; Bickhard & Terveen, 1995), has identified a deep problem with assumptions held by current computational approaches to representation. These problems are part of the cause of the above fragmented approach to the study of cognition. His alternative approach to representation, *interactivism*, avoids these mistaken assumptions and also accords with situated cognition's central tenets of modeling cognition from the perspective of autonomous systems in interaction with the world. This alternative is inherently developmental and presents the foundation for how representation emerges out of the functional organization of a system in interaction with an environment. However, we still lack an account of the kind of "structure" that seems to be required for explanations of phenomena like analogy. My goal is to take the first step towards developing an account of representational structure based on interactive representation.

The present dissertation is thus split into two well-defined parts. The first is devoted to presenting a clear statement of the problem: uncovering what exactly is lacking in the current approach to structured representation, what it is about this approach that entails such shortcomings, and what, then, needs to be re-evaluated. I do so by developing a methodological tool designed to expose the shortcomings of the current approach to computational cognitive modeling which assumes the unproblematic presence of structured representational content. This tool is the *framework for the identification of handcoding*. Handcoding is the inappropriate involvement of humans in the modeling of cognitive phenomena, which results in leaving unexplained aspects of the phenomena which the model was intended to explain.

I use current research on analogical cognition as a case study to focus and ground my investigation because this research has been very successful in explaining a variety of properties of analogical cognition and also has a large amount of established dialogue to work with concerning handcoding issues. At the same time, this research also has clear

gaps in its explanatory power precisely because of the effects of the handcoding of structured representations in current models. The result of this handcoding is that we have an incomplete view of the central nature of analogical cognition and its role in the underlying ontology and development of concepts in autonomous systems. What is needed is an account of the *emergence* of representational structure from a non-handcoded perspective.

The second part begins by explaining Bickhard's interactive representation framework and its departure from the general approach assumed in the models I have investigated. This lays the foundation for an account of how representation emerges and changes in a situated, autonomous and interactive system. I then augment Bickhard's interactive representation framework to develop the framework for *situated representations*: an account of how structured representations based on interactivism can emerge through situated interaction with the environment. The result is an account of structured representation emergence grounded in interactivism. And a corollary is that representation is only properly understood in the context of this interactive account. This dissertation will *not* provide a complete new model of analogical cognition. However, it will demonstrate that the kind of representational structure required for such an account is achievable within the situated representation framework.

The significance of this project is that it proposes a reorientation of approaches to representation that opens up new avenues for research and modeling possibilities previously not available (e.g., accounting for representational content emergence in computational models). The project takes a first step in reconciling the tension between accounts which currently require structured representations and the recognition that a story must be told of how such representations get there in the context of a history of situated interaction. It also clarifies the utility of symbolic representations and their role in scientific investigation; while some explanatory goals cannot be attained within the symbolic representation framework, they are nonetheless an essential part of psychological investigation (as evidenced by current progress). This dissertation therefore speaks to a number of different research venues within cognitive science, including methodological issues in computational cognitive modeling, research on analogical cognition, situated cognition, and the fundamental nature of representation; the project also has implications for issues regarding the symbol-grounding problem, the problem of intentionality, cognitive development, and the semantics of mental states.

I turn now to set the stage for the subsequent chapters and to introduce the problem that I will solve. This requires outlining in more detail the motivation for an account of situated representations, taken from the perspective of explaining analogical cognition. It was from this perspective that I was led to the discovery of a need for situated representations. Following this account, I briefly outline the order of the content in the subsequent chapters of the dissertation.

The situated representation project: Illustrating the central problem and its solution

I come to this discussion carrying a theory of analogy which cannot be adequately described by current computational accounts of analogical cognition (Dietrich, 1996, in press; Dietrich *et al.*, 1996; Morrison & Dietrich, 1995; Oshima, 1996). This is not to say

that this theory is strictly at odds with current psychological models; in fact, it is my hope and belief that this project will turn out to complement and extend what has currently been learned from such investigations, and make it possible to address issues that until now have been inaccessible to computational models of analogy. The problem, rather, is that the current computational approaches cannot, in principle, capture what I believe is integrally tied to the central nature of analogy: *the capacity for the emergence and change of fundamentally new ways of representing*. The intuition here is that analogy, at its core, is a creative process in which fundamentally new ways of representing the world are produced as a result of analogical cognitive processing. In the language of current symbolic approaches, analogy depends upon or results in the emergence of *new representational primitives*.

It is important to note here that the mechanism for the emergence of new representation is not a central issue just to analogy — it is a necessary condition for learning and developmental phenomena more generally: i.e., any cognitive phenomenon that involves the emergence, extension or change of representations in a cognitive agent. The framework for representation that I pursue should be applicable to learning and development writ-large. Focussing on analogical cognition, however, is ideal for two reasons. First, it helps constrain the focus of the project to a particular phenomenon; learning in general is far too broad, and keeping the discussion focussed on a more specific phenomenon helps to make clear what an actual theoretical result would be. Second, analogy is particularly interesting because, while it appears to be present and play an important role very early on in cognitive development, it also requires representations which have some sort of structure to allow for organization, comparison, and combination so that analogical comparisons can be made — in fact, it is proposed that the developmental appearance of analogy-making at all is the result of the development of structured representation (Gentner *et al.*, 1995). It is precisely this presence of structure, while at the same time not requiring overwhelming complexity in structure, that makes accounting for the emergence of the kind of representation required for analogical cognition an ideal target.

Finally, it is also important to make explicit the logical dependency of the claims which I am making. My claims about representation are, in an important sense, logically prior to my claims about how representation works in analogy. I was led to a need for developing the situated representation framework because of a desire to account for aspects of analogical cognition which cannot be made in present approaches. However, it is possible for my claim about the integral role of foundational representation emergence and change *within* analogical mechanisms to turn out to be false, while the need for the situated representation framework remains. There are two possible stories about the relation between analogical cognition and the emergence of representation: (1) the emergence of representation is involved in *how* analogies are made, and therefore analogy mechanisms cannot be separated from these learning mechanisms; or (2) analogy mechanisms are separable from representation emergence — nonetheless, representation emergence has to happen *somewhere*, and analogies are made based on these representations. In either situation, demonstrating the lack of representation emergence in current computational models of analogy is enlightening: it highlights the kinds of representations which we need for an account of structured representation while also exposing how current accounts capture many of these features *without explaining where*

they come from. In this way, I am making two distinct claims: (1) about how representation must work, and (2) about the involvement of representation development in analogy. The first is the central focus of this dissertation. The second, I also hold true, but defend only partially; a full test and defense of the second claim must wait until a fully developed model of analogy based on the situated representation framework exists.

(It follows that demonstrating the lack of representation emergence in current models of analogy has two interpretations: (1) lack of emergence entails that we do not yet understand analogical cognition; or (2) lack of emergence entails that we do not yet understand the fundamental nature of representation in learning, which analogical mechanisms closely depend upon. I hold that (1) is likely, and (2) is true.)

Representation and analogy

So, what does novel representation emergence and fundamental representation change mean in terms of analogy, and how is it lacking in current computational accounts? Analogical cognition, and the ability to perceive similarities in general, is believed to play a fundamental role in many cognitive abilities. As Vosniadou & Ortony (1989) point out, it is generally believed that the capacity for recognition, classification, learning and creativity stems from the ability to perceive similarities and analogies. In particular, the cognitive agent that makes an analogy extends and changes its currently represented concepts: it learns that one thing *is like* another¹ – and this learned relation was neither understood nor represented before the similarity comparison was made. Furthermore, this is not a matter of simply matching antecedently identical but not yet compared aspects of a concept represented in the head. Rather, this is a case of full-blown representational content emergence. I claim that a computational explanation of analogy must account for the change from “no prior existing representation of two things being alike,” to “representing that one thing *is like* another in some way.”

Take, for example, a situation of analogical reminding, where some experience reminds a person of another object or situation: a man walking down a street late at night sees a jumbled pile of garbage cans strewn in a driveway and is immediately reminded of Stonehenge. Something about the arrangement of the garbage cans prompted him to think of Stonehenge; in some way, for the man, the garbage can arrangement *is like* Stonehenge. Prior to the analogical reminding, the man probably never considered garbage cans as possibly resembling Stonehenge. Furthermore, the observation that the garbage can formation is like Stonehenge highlights for the observer an aspect of Stonehenge not seen before: the abstract arrangement of the monoliths in a circular, but

¹ It is important to note that a recognition that two things are alike is associated with its compliment: that they are also *unlike* one another, and in ways different from those that make them similar. Thus, more information is involved (and becomes accessible) than merely the positive side of similarity — anywhere there is similarity without identity, there is also dissimilarity. In fact, in ancient Greek thought, analogy was contrasted with *polarity*, the perception of differences or opposites, which was expressed in the same format as analogy (Lloyd, 1966; Hoffman, 1995). I will bring this issue up again, below, and highlight its significance. For the present discussion, take my use of the *is like* comparison to include its compliment: “*is also unlike*, and in different ways.” (It is also important to note that these differences, and to an extent with similarities as well, may be implicit, as opposed to explicitly or consciously represented — e.g., they may require some sort of inference in order to become explicitly represented).

characteristically jumbled pattern. This too was not antecedently represented, at least in a highlighted sense, in the man's concept of Stonehenge.² Conceptual change has occurred, resulting in a new representation of the situation and the man's associated concepts (including change in recalled concepts that he had prior to the analogy). Furthermore, I believe that there is the possibility, even in analogical reminding, for there to be the production of the representational capacity for a novel kind of similarity relation or category that was not previously represented prior to the reminding.³

The problem: handcoding representational content emergence and change

While current models of analogy, including Dedre Gentner's Structure-Mapping Theory (hereafter, SMT; Gentner, 1983, 1989) and Douglas Hofstadter's theory of High-Level Perception (hereafter, HLP; Chalmers *et al.*, 1992, Hofstadter, 1995), have illustrated important computational properties of analogical processes (such as the roles of perception and systematicity in the processes of comparison of structured concepts), they have missed a full account of how the representational repertoire of the cognitive agent is changed and extended — and in this sense they have missed one of the hallmark features of analogy. Of course, current models do account for aspects of representational change and re-organization (Gentner & Wolff, in press), but the key claim is that they do not account for the possibility of the emergence of a fundamentally new way of representing — the learning of a new kind of relation or category; each of the current models fails to account for the emergence of fundamentally new representations based on similarity and difference relations between the subjects of comparison.

Nonetheless, these models appear to make analogies of the kind that humans make. Accounting for why these models look as if they're making the full gamut of analogies, yet fail to account for the potential emergence of representation of a new category, requires developing a framework for the identification of a potential problem with computational modeling (and scientific models more generally): *the handcoding problem*. Handcoding in general concerns the involvement of humans in the operation of a computer program in such a way that the behaviors of the program judged as “genuinely intelligent,” “creative,” or “unique” can only really be said to be a direct product of the programmer, rather than a novel production of the program itself. Of course, handcoding in general is a natural and necessary part of making a model and using it as an explanatory device. However, I will make a distinction between handcoding that is legitimate and handcoding that is illegitimate. This legitimacy depends on what it is the

² The argument here is that it is simply implausible that *every* possible association, structure, detail, etc., or combination thereof, is represented antecedently. Dietrich (in press) presents and explores this as *the Low Probability Argument*. Camac & Glucksberg (1984) also present empirical evidence suggesting that there are *no* associations between certain concepts prior to metaphorical comparisons that subsequently result in associations.

³ This is the deepest assumption I make about analogical cognition and is, I believe, both an empirical issue (it is possible, but we (Dietrich, in press; Dietrich *et al.*, 1996; Morrison & Lee, 1998) believe highly unlikely, that concepts are structured such that they are semantically “close-enough” to antecedently match), and a metaphysical issue (avoiding handcoding of representation logically requires emergence of representation; discussion of encodingist versus interactive approaches to representation in Chapter 4 will deal with this in more detail).

model is attempting to explain.

It is my contention that current models of analogical cognition have handcoded their representations in such a way that they lack (necessarily) an explanation of how novel representation emergence and fundamental representation change occurs — and therefore, according to my hypothesis concerning the importance of such processes in analogy, lack a full explanation of analogical cognition.

The researchers of SMT and HLP have handcoded the representations in their models so that the account of how representation emerges and changes as a result of analogical comparison is bypassed. In each case, a human is required to set the programs up in such a way that the emergence of a fundamentally new category or *is like* relation (or change of a fundamental way of representing) in a comparison of subjects is not necessary: the human has already taken care of setting up such representations. The SMT and HLP models don't explain how such emergence and change occurs, as this process still exists only in the human: the SMT and HLP *researchers* are really the ones responsible for the initial representation construction. This reduces the rest of the analogy process in their models to a simple identity matching (and this is true whether the identical structures already exist explicitly (as atomic but inter-related symbols), or are to be constructed, with the identity relations implicit in the unchanging construction rules antecedently pre-set by the researcher).

This problem becomes particularly clear in the face of developmental data that strongly suggests that such emergence and change is possible, not only over developmental time, but within single experimental runs where the only pressure is the need to make kinds of relation-based comparisons (Barsalou, 1983; Gentner *et al.*, 1995; Morrison & Lee, 1998).

I argue that the lack of an account of representation emergence and change in analogy-making misses the core of the analogical-comparison process because the greater part of the computational work in engaging in an analogical comparison is in the process of discovering and sustaining how two antecedently non-identical situations can actually be treated identically, which potentially involves constructing novel categories and changing existing capacity to represent. An account of representational content emergence and change is required to fully explain the capacity to make these novel similarity comparisons.

While current models have relied on this kind of handcoding in order to dodge some of the deep issues regarding the nature of representation and its relation to analogical cognition, it is very important to emphasize that such handcoding has not been devastating to accounts of *all* aspects of analogy. In fact, tremendous advances have been made, and will continue to be made, while still relying on the handcoded representation scheme assumed by current models. However, solving this handcoding problem with respect to representation is central to a full understanding of representational processes as they occur in representation-based cognition (such as analogy) — including how representation plays a role in learning and development. Thus, exposing this handcoding serves to demonstrate one place where this new view of representation can do much work; at the same time, it is also clear that this new view of representation must be able to account for the phenomena that our best theories have uncovered about how analogy works (i.e., the kinds of features of representation that analogical cognition requires).

Avoiding handcoding: situated cognition and interactive representation

Removing handcoding from a computational model often requires reworking the entire model. With respect to accounting for representation change and emergence in analogy, a model is required that is quite different from those currently proposed. In order to make this account while avoiding handcoding, an obvious methodological path to take is one that removes as much of the researcher's involvement in the computational processes as possible. I argue that the best methodology to adopt in order to avoid the kind of handcoding with respect to representation construction found in current analogy models is that proposed by *situated cognition* (also called *situated action* or *situated robotics*). The core thesis of situated cognition, the thesis of *embodied cognition*, holds that cognition is best understood from the perspective of autonomous, active agents situated in an environment with which they interact — a perspective which, it is argued, forces us to consider the internal and external functioning of a system ecologically, removing as much as possible our own experiential biases as to how the agent perceives, cogitates and acts within its own perspective of the world. A situated explanation of a cognitive capacity is thus a matter of analyzing the relation between the internal workings of the agent and its performance, over time, in interaction with the environment towards some goal (Agre, 1995; Clancey, 1997; Hendriks-Jansen, 1994, 1996; Loren *et al.*, in press).

Forcing the model to be a situated and autonomous agent promises to remove much of the possibility of handcoding: if we want our agent to be adaptive (function intelligently) on its own in the world, its representations can not be pre-defined and static, but will have to evolve over time in the service of the agent's goal-directed activity. The agent itself has to maintain the connection between its internal indications of what to do and the external states of affairs, and has to face the consequences of its actions — the experimenter is not allowed to play the role of buffer between the “internal cognitive world” and the external world. Being an autonomous agent (at least in theory) thus entails that the agent has to solve its own epistemic problems. Proponents of situated cognition, however, have often eschewed the classical notions of concepts and representations (Brooks, 1991). This is a problem, as I hold that the notion of representational (conceptual) structure is still necessary in an explanation of analogy, as well as other high-level cognitive phenomena. This rejection has also been the source of much of the scepticism surrounding the situated cognition approach, and has led to charges that it will not be able to “scale-up” to higher-level cognition. This has been referred to as the *scaling problem* (Kirsh, 1991; Tsotsos, 1995).

Furthermore, the cognitive psychologist as analogy-researcher might rightfully ask: What does ongoing interaction with the world have to do with the “internal” comparison of concepts, except perhaps to “ground” those concepts and make them refer? — Why is such “hook-up” necessary for internal comparison? The short answer is that it is only while situated in the context of functionally maintained interactive competence with an environment will representation emergence and change be possible (although the ‘environment’ is not necessarily “the world ‘out there’”). This challenge in fact cuts to the core thesis of this dissertation and will be fully explored in the last chapter.

The scaling problem presents us with the tension described above. On the one hand, handcoding is certainly something that must be avoided in order to save our explanations in computational models — for making certain that *we* (the builders, operators and interpreters) aren't solving problems that our computational models should be solving on their own. And the best way to eliminate handcoding with certainty is to take the situated cognition perspective and methodology. At the same time, however, situated cognition appears to eliminate representations, the very explanatory entity we need for accounts of cognitive phenomena like analogical cognition.

One possible route to take, given this seeming dilemma, is to deny that representations exist in any form; thus, cognitive phenomena like analogical cognition will require a very different kind of explanation (Thelen & Smith, 1994; van Gelder & Port, 1995; Wheeler, 1996). I believe, however, that this is the wrong approach. I argue that these dismissals of representation are a result of lacking an alternative approach to representation and do not constitute an in principle denial of all forms of representation. One such alternative approach to representation is offered in Mark Bickhard's *Interactive* framework (Bickhard, 1980a, b, 1992a, b, 1993; Bickhard & Richie, 1983; Bickhard & Terveen, 1995; Campbell & Bickhard, 1986), and I argue that it will break the tension posed by situated cognition's methodological solution to handcoding.

Interactivism is a response to what Bickhard calls an *encodingist* approach to representation — an approach that is fundamentally incoherent, and thus is to be rejected. Encodingism is based on the premise that a fundamental form of representation is one in which representations have “contents” (which are treated as objects), such that representations are like containers that may be manipulated without changing their “contents” (the manipulation point can be replaced by activation of connectionist “categories” and still retain its force; Bickhard & Terveen, 1995). Furthermore, these contents represent what they do — make the representations be *about* something in particular — in virtue of “standing in” for what is to be represented. Bickhard's extensive exploration and development of his critique of encodingism demonstrates that most of the current approaches to representation explicitly or implicitly assume an encodingist position (Bickhard, 1993; Bickhard & Terveen, 1995). Encodingist approaches, however, are based on a logical incoherency: there is no possibility of accounting for how the contents of such encodings arise, how they stand-in for what they do, or what they are of, within the framework of encodingism itself — the only recourse to explaining content within encodingism is the vacuous claim that, “they stand in for whatever it is that they stand for.”

I demonstrate that the approach to representation assumed by the SMT and HLP models can easily be interpreted as a species of encodingism (and attempts to ground the approach in a non-encodingist framework do not seem empirically viable). I then show how encodingism, in turn, entails necessary reliance on the handcoding of the representations — both in how they get there and how they are interpreted (either by observers or the system itself). Thus, avoiding encodingism entails avoiding necessary reliance on handcoding representational content, opening the way to the possibility of a non-handcoded account of representational content emergence and change.

Bickhard's alternative to encodingism, *interactivism*, is founded on the premise that representation is inherent in functional organizations of behavioral control structures that embody the capacity to develop and maintain successful interaction with the environment

in the service of a (system-internal) goal. Representational content in these control structures is determined by what further possibilities for interaction are indicated by the activation by the environment of certain internal states. How the system makes contact with the world is via the internal system states that the system ends up in as a result of a particular interaction with the world. These internal states implicitly define classes of environmental stimuli, but importantly, the agent does not know what it is that activates them simply in virtue of their activation. Instead, these internal states may play a representational role for the system by being linked with behavioral control structures so that while an internal state is active, it indicates that *if* a certain action (or more precisely, activation of an action-generating subsystem) is initiated, certain other internal states should become activated. The indicated outcomes, in turn, may indicate other further actions with outcomes — interactions. Actual outcomes then provide the system with information about whether its action was appropriate, given what it expected. In this way, the root level of representing the world — of the system having some idea about what the world is like — is based on these indications of potential further interactions; and how the world actually is, is discovered through the experience of the agent interacting with the world.

This perspective avoids the impasse of encodingism because these representations may be built out of initially non-representational functional system-organization — there is no necessary reliance on a presupposition of already representational system states that somehow inform the system of what they are about in virtue of their activation by some external condition of the environment. Furthermore, in the service of goal-directed selection of behavior, the interactive conception affords the possibility of system-detectable error — a necessary condition for learning: when the system's action does not result in what it expects, this indicates to the system that it is not functionally organized appropriately to predict the way the world is so that it can achieve its goal.

Certainly, this root level of representation is far from the kinds of structured representations required for analogy. But I argue that there is good reason to believe in the possibility of this account scaling to such complexity — and to do so based on learning from a history of interaction, and not from pre-established representational primitives. I demonstrate this by augmenting the interactive representation framework; the key suggestion that I take from Bickhard's work is that through a history of interaction with the environment, expectation-based representations begin to develop their own internal structure based on patterns of linked expectations, and these webs of linked expectations may develop their own distinct properties — properties which are originally derived from the agent's species-specific interactions with features of the world. These representational properties, in turn, may become the objects of interactive expectations, thus resulting in new expectations — new represented categories — of interaction outcomes with the environment. Likewise, existing expectations have the potential of being modified — changed — on the basis of interaction outcomes: if the expectation is not met, the agent has the usable information that it was mistaken about the appropriateness of the action just taken to reach its current goal; the agent may subsequently change its expectation of future outcomes based on this experience (e.g., to expect in future similar interactions that which actually resulted in this current interactive context). This, I argue, is the first step towards representation structure which can emerge and change on the basis of a history of interaction with the environment. I

develop this account in detail in an artificial life thought experiment.

Much further work is required to attain the robust kind of representing of relations, entities and attributes of the world which current analogy models claim to work with; likewise, we still lack a full account of the kinds of representational dynamics that would result in analogical comparisons of existing representation structures (e.g., how internal interaction of representation structures can be sustained, leading to a variation of interactive emergence similar to direct interaction with the environment). Nonetheless, demonstrating the potential for emergence of subsequent higher-order representing on the basis of lower-order representational properties does complete my project at hand, which is to provide a non-handcoded account of how representational content can emerge and change, and how structure can emerge out of organization of such content.

The dissertation chapters

The chapters are arranged as follows. In Chapter 1, I introduce the problem of handcoding in general, describing its roots in problems concerning creativity, autonomy and explanation. In Chapter 2, I develop the framework for the identification of handcoding in computational cognitive models, explaining how handcoding affects the logic of explanation. In Chapter 3, I lay out the problem of handcoding with respect to the emergence and change of novel representational content in current models of analogy. This serves to demonstrate current model shortcomings and how they cannot solve them within their current assumptions, as well as make clear the kinds of structures that will still be required in an alternative account. Additionally, I discuss developmental data that strongly suggests that representation development is very closely coupled with (and perhaps a part of) analogical comparison mechanisms. In Chapter 4, I demonstrate that present handcoding is a result of reliance on encodingism, and describe Bickhard's interactive solution to the encodingism impasse to explaining representational content and its emergence. I provide a detailed thought experiment of an artificial life form that shows how situated representations emerge. Finally, I indicate the path we should take if we are to implement systems in which relational representation emerges.

A diagrammatic tour

The diagram below (Figure I.1) depicts the subject-matter of the following chapters from the perspective of a critical analysis of the representational approach assumed in current models of analogical cognition, raising key problems in the foundation, and providing an alternate approach to avoid the problems, including proposing how the modeling of representation involvement in analogical cognition can be built back up. Metaphorically, Chapter 1 argues that there is a picture to be drawn, Chapter 2 gives us the tools to draw it, Chapter 3 draws the upper half, and Chapter 4 passes over the boundary and sketches how the lower half should look.

Dissertation Territory

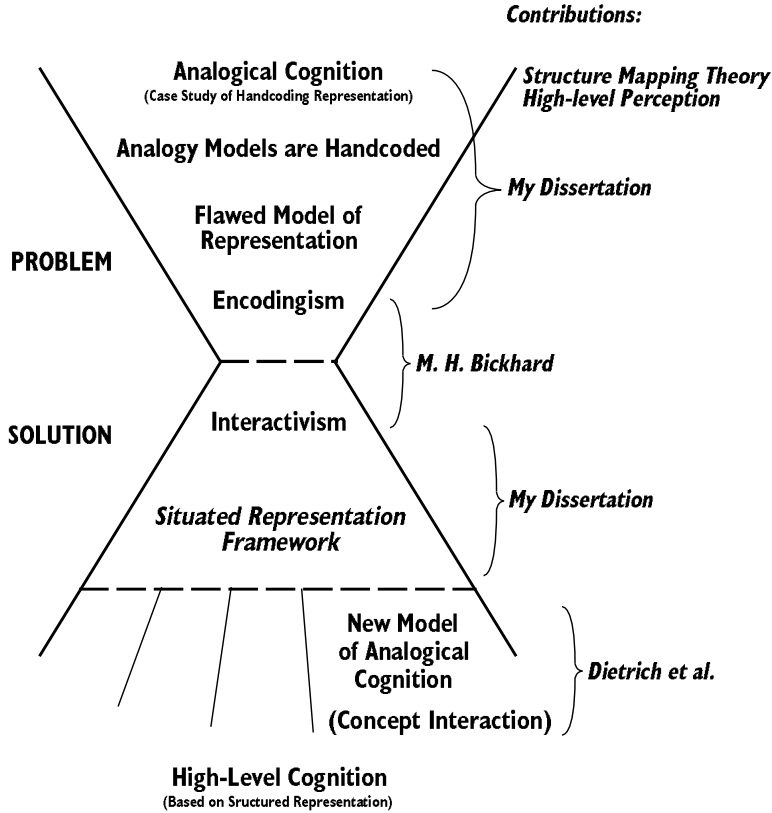


Figure I.1 - Diagrammatic tour of the dissertation territory

“ Oh, mice have died, and worms have eaten them; but no rock, and no spiral nebula — and no worm, for that matter — has ever chased a mouse, let alone caught one. (Mousetraps catch mice, of course; but that manifests our intelligence, not theirs.)”

J. Fodor, *Psychosemantics* (1987)

Chapter 1

The Handcoding Problem

1 - Introduction: The Challenge of Computational Creativity

In the mid-nineteenth century, Lady Ada Lovelace presented the first published argument against the possibility of a computer being independently creative. Charles Babbage, Lovelace’s close friend, had completed what is now considered to be one of the first designs for a digital computer. Babbage named his design the ‘Analytical Engine’. Lovelace and Babbage corresponded with one another and considered the possible capabilities of the device, including its capacity to “compose elaborate and scientific pieces of music of any degree of complexity and extent” (Boden, 1991, p.6). Although Lovelace agreed that the potential capacity for complexity of the Engine was, in principle, unbounded, she concluded that,

“The Analytical Engine has no pretensions whatever to *originate* anything. It can do [only] whatever we know how to order it to perform.”

(Lovelace, in Bowden, 1953, p.398)

That is,

“Any elaborate pieces of music emanating from the Analytical Engine would therefore be credited not to the engine, but to the engineer”

(Boden, 1991, p.6).⁴

⁴ Boden (1991) actually distinguishes between four “Lovelace” questions: (1) Can computational concepts help us to understand human creativity? (2) Could a computer, now or in the future, appear to be creative? (3) Could a computer, now or in the future, appear to recognize creativity? And (4) could a computer, however impressive its performance, really be creative? Boden notes that it is possible to answer yes to the first three without necessarily answering yes to the fourth.

The goal of getting a computer to produce original and autonomous intelligent behavior has remained one of the central concerns for artificial intelligence (AI). Arthur Samuel voiced a challenge similar to Lovelace's in the 1950's, at the time when AI was being founded as a science:

“How can computers learn to solve problems without being explicitly programmed? In other words, how can computers be made to do what is needed to be done, without being told exactly how to do it?”

(Koza 1992, p.1)

This challenge is a natural one for AI to face because AI is concerned with the production of intelligent systems, and intelligence is generally held to be the capacity to autonomously perform complex tasks. This goal has produced a variety of methodological concerns commonly expressed in criticisms of computational research projects, such as discouraging the use of *ad hoc* programming structures in order to achieve some “naturally produced” behavior by a program. These sentiments are particularly echoed in the branch of AI that is part of general cognitive science, where the purpose of computational models is not only to produce behavior, but to *explain* how intelligent phenomena arise in already existing, naturally intelligent systems.

In this chapter, I will take up this issue of autonomy and explanation in computational cognitive modeling under the general notion of the *handcoding problem*. One of the most colorful contemporary expressions of versions of handcoding has been given by Douglas Hofstadter in his book, *Fluid Concepts and Creative Analogies* (1995). In fact, Hofstadter first introduced his notion of “handcoding” in the 1992 paper, *High-level perception, representation, and analogy: A critique of artificial intelligence methodology*, co-authored with David Chalmers and Robert French (Chalmers *et al.*, 1992). The notion as they use it, however, while highlighting aspects of the problem, is inadequate for capturing all the issues involved. I wish to strengthen the notion of handcoding by relativising it and outlining the proper use of its identification as a methodological tool in scientific investigation.

To get a grip on what handcoding is, I have started with the historical roots of the concern over handcoding with respect to “creativity,” as voiced by Lady Lovelace. Creativity is often taken as being one of the distinguishing features of being human and intelligent. This makes creativity a good place to start because it is a phenomenon that is a product of an intelligent agent (what cognitive science is attempting to explain), and we all have strong intuitions about what is distinctly creative and what is not: when something that was deemed as a “creative act” is exposed as not being creative in the sense we had originally thought, we are usually very satisfied that we have exposed some fraud. “Creativity,” per se, however, is not the only phenomenon that concerns issues of potential handcoding in models, as will become clear. There are many aspects of cognitive phenomena that may be handcoded, and I will be developing a framework for its identification that is applicable to potentially all conditions in computational cognitive modeling which could involve handcoding. First, however, I will introduce the handcoding problem by giving an intuitive definition and some clear examples in AI and cognitive science where handcoding has been previously identified.

2 - A first-pass definition and an intuitive example of “Handcoding”

A first-pass definition of handcoding is that it concerns the involvement of humans in the operation of a computer program in such a way that the behaviors of the program judged as “genuinely intelligent,” “creative,” or “unique” can only really be said to be a direct product of the programmer, rather than a novel production of the program itself (hence, the program’s intelligent or creative behavior has been “hand-coded” by the programmer).

Before unpacking the many issues wrapped up in such a notion I will first distinguish handcoding from what it is not. Namely, I do not hold the opinion that just because a human designs a computer program that the program itself, in principle, cannot be said to have the capacity for genuine, independently intelligent and novel behavior. As Boden (1991) correctly points out, while Lovelace was correct in asserting that a computer can only do what its program enables it to do, it does *not* follow that there can be no interesting relations between creativity and computers. (The same holds for other manifestations of cognitive phenomena besides “creative acts.”) One way to put the argument is that we don’t yet know how to order the computer to do things in the kind of novel and creative way that we do (again, similar points could be made for other cognitive accomplishments beyond distinctly creative ones). Thus, prior to the question of whether there is an in-principle limitation which would deny any full computational account of creativity, lies the question of what creativity itself is.

I cannot hope to answer this prior question, however, without having what would probably amount to a whole theory of mind. And it is unlikely that a whole theory of mind will be in the offing for quite some time. Nonetheless, I do believe that a relativised notion of handcoding can provide a useful methodological tool to enable us to take steps towards creating computational models whose capacities and behaviors go increasingly beyond the given set of behaviors recognized as a direct product of the program’s intelligent human creators, and on towards novel and adaptive autonomous behaviors. I argue that such a notion of handcoding would provide us with a comparative measure of how “independent” a program is. And this relative independence is a partial measure of how autonomously intelligent that program therefore is.

Just because creativity is a profound issue does not mean that we don’t have *any* idea of what it is — or when something fails to demonstrate creativity or novelty. The cases in which we might deem some action as creative or novel can be roughly defined as: cases in which the information present in the environment does not explicitly suggest some inference (or in cases where the agent is currently in a situation in which it perceives a problem, but a solution is not explicitly represented), yet, through some process, such an inference is made (or a solution to the perceived problem is found).⁵ Starting with this idea, handcoding can be described as occurring in cases where there

⁵ “Novelty” and “creativity” are usually taken as matters of degree — *how novel* something is, or *how much creativity* was involved in its production. This measure of degree is generally a factor of “how far beyond the information given” the process goes, as well as “how appropriate” the answer is to the problem (also, often, how “simple” or “elegant” the answer is). Bruner (1957) was one of the first to discuss intelligence in terms of “going beyond the information given.”

should have been a process that “calculated” the answer, but by some set of circumstances, the “creative inference” or “answer” was instead “handed” to the agent — the answer being a product of some external process (e.g., a human programmer) which the agent did not itself have or produce.

2.1 - An Intuitive Example of Handcoding

So, what might handcoding look like? The following thought-experiment will introduce the idea. Suppose you were asked to judge three programs with respect to whether they were good models of how average high-school educated people multiply.⁶ You first test the programs by giving them several pairs of positive integers to multiply (suppose you choose integers less than 100) and see if the programs produce the correct answers. After a number of test instances, you are satisfied that they can multiply. Now you look at the code of each program to see *how* each arrived at its answers.

Examining **Program 1**, you find that it is actually a very simple program: it consists of a large look-up table by which pairs of integers are equated with a third integer, which is output as the answer. The table expands to cover any combination of two integers up to a cardinality of 100 each, in any combination (e.g., $\langle 2,3 \rangle = 6$, $\langle 4,5 \rangle = 20$, $\langle 27,3 \rangle = 81$, etc...). On the face of it, the explanation this program offers as to how people multiply seems rather unrealistic. To begin with, it’s unlikely that people have memorized the answers paired with any possible combination of two input integers up to 100 each. Most of us have probably memorized “times-tables” for integers up to 10 each (possibly more, based on experience or depending on what we were required to learn in grade school); but much beyond that is likely rare, at best.

But an even bigger problem is that this program is *brittle*. Suppose we wanted to multiply 101 by 1. This entry doesn’t appear anywhere in the table — the program cannot handle it. But anyone familiar with multiplication would be able to instantly give the answer. This program seems to fail as a good description of how multiplication in humans works — in fact, it fails as even an instance of the capacity to multiply *at all* if we multiply integers over 100.

⁶ Two caveats should be made here. First, multiplication is not a terribly “creative” process. A better example of actual creativity might be to consider a situation in which some people are asked to come up with a jingle for a radio commercial of some product — here the conditions are such that there is not a clear rule-based decision to be made, so the space of possibilities is much greater, and therefore “good” results would seem to require much more “creative” thinking. However, the multiplication example is an instance where we can at least say that *before* the numbers are multiplied, the agent doing the multiplication does not have the answer explicitly, but *after* the multiplication process is over with, the agent does have some answer. The multiplication example should therefore be enough to demonstrate to what extent we might judge that “the answer has been unfairly given to the program,” or not.

Second, there is most likely not just “one way” of doing multiplication — in a given population, there may be a variety of different techniques for getting the “right” answer, some perhaps holding nothing in common with the others except that they arrive at the same answer. Nonetheless, we can still make judgements about whether some process is a real instance of improper handcoding such that the task of multiplication isn’t really necessary since the answer has already been provided.

The most striking feature for my purposes, however, is that *all* of the possible answers for multiplication of integers of 100 or less *exist antecedently* to an integer-pair being presented. And the program relies entirely on the pairings in the look-up table for its correctness as an answer to the multiplication of two integers of 100 or less. The program itself plays no part in the process in which integer-pairs are paired with certain answers — such pairing was entirely the product of the programmer. In this sense, the program doesn't really multiply. Rather, the programmer *multiplied* and made a table of answers, which were then coded up in the program. Because of this, the program has been handcoded.

Program 2 is a little more complicated. First, it checks to see if either of the integers are zero. If they are, it returns a zero as the answer and halts. If both integers are greater than zero, then it goes into the following loop: first, it takes the cardinality of the second integer and adds it to a new set, Integer Number 3 (which initially starts out empty), and then takes the cardinality of the first integer and subtracts one from that integer. It then repeats this process, adding the whole cardinality of the second integer to the third set again, and subtracts one more from the first integer (which is already one less from what it started with because of the first subtraction). And it repeats this process in a loop until the value of the cardinality of the first integer is 0. At this time, the newly created set, Integer Number 3, is presented as the answer and the program halts. This algorithm in Program 2 is called the *Repeated Addition Method* (RAM).

The major advantage of this program over the first is that it is extensible to any size of multiplication (ignoring memory constraints on sizes of integers). No matter how large the two integers being multiplied are, this program, in principle, could calculate the answer through its method of repeated addition. And, unlike the first program, that answer does not have to antecedently exist in the program; instead, the program has the procedure to calculate the answer given any two positive integers as input. Handcoding of the kind described in Program 1 isn't present in Program 2.

There are still some drawbacks with this program, however. The biggest drawback is that when the numbers get very large, repeated addition, while not too much of a problem for a computer, would be quite tedious for a human to count through. It would certainly be difficult to keep repeated additions in memory if the integers were over 1000 each; and even keeping track of the amounts on paper would become both time and space consuming.

Program 3 is even more complicated than Program 2. Rather than relying on repeated addition through the whole integers, it instead uses the RAM algorithm of Program 2 for only small portions of the calculation. In fact, the procedure used in Program 3 turns out to be the one usually taught in elementary mathematics, called the *Partial Products Method* (PPM). The procedure itself is actually rather simple, but explaining it is best shown in an example so as to account for all the different variations that might occur in different problems.

Suppose we gave Program 3 the following two integers as input: 3482,24563. The program first identifies which input integer is the largest, and labels it the *Larger Integer*; the other integer is then the *Smaller Integer*. The program then breaks up the Larger and Smaller Integers into sets of values between 0 and 9 with respect to their "place" within a decimal representation of the Integers: these numbers are the decimal representation of the constituent parts of the Integers with respect to factors of 10; so, e.g., in the decimal

representation of the integer 3482, the 4 is in the 100's place. The two Integers now have the following representation:

$$\begin{array}{r}
 \begin{array}{c}
 10000's \\
 1000's \\
 100's \\
 10's \\
 1's
 \end{array} \\
 \begin{array}{r}
 24563 \\
 \times 3482 \\
 \hline
 \end{array}
 \begin{array}{l}
 \text{Larger Integer} \\
 \text{Smaller Integer}
 \end{array}
 \end{array}$$

Figure 1.1 - Representation of the Problem

The next step is to go through the basic Loop of the program (Figure 1.2). The first step in the Loop (Step A) is to take the one's place of the Smaller Integer and multiply it (using the RAM algorithm) by the 1's place of the Larger Integer. The total product of this multiplication may be greater than 10, but will be equal or less than 81 — so there is the possibility of a 10's place, but not a 100's place in the decimal representation of the product. Whether the product is greater than 10 or not, the 1's place of that integer will be added to a *Third Integer*. In this first run through the loop, whenever an amount is added to the Third Integer, it will always be to the place that is the same as the current place of the Large Integer being multiplied (in this first step, the 1's place); this place grows in magnitudes of 10 with subsequent runs through the loop, which I'll show below. Thus, in the example, the first product is 6, which is less than 10, so this is simply added to the first place (i.e., the 1's place, because we just multiplied the 1's place of the Larger Integer) of the Third Integer (Figure 1.2, A). The Third Integer will act as the storage for the eventual answer.

The next step in the run through the Loop for our example (Step B) is to multiply the 1's place of Smaller Integer by the 10's place of the Larger Integer (Again, using RAM; Figure 1.2, B). The product of this multiplication is 12, so now there is a 1's and a 10's place in the product. The 1's place of the product, 2, is added to the next place in the Third Integer (in this case, the 10's place because the current step is at the 10's place of the Larger Integer), and the 10's place of the product, 1, is kept to be added later to the 1's place of the product in the next step of the Loop — this is commonly referred to as the *carried number* (in Figure 1.2, B, it is represented as the small 1 above the 5 of the Larger Integer). The third step (Step C) now multiplies the 1's place of the Smaller Integer with the 100's place of the Larger Integer: 5 (Figure 1.2, C). Their product is 10, so again we have the situation like Step B: the 0 is in the 1's place of the product, and the 1 is in the 10's place of the product.

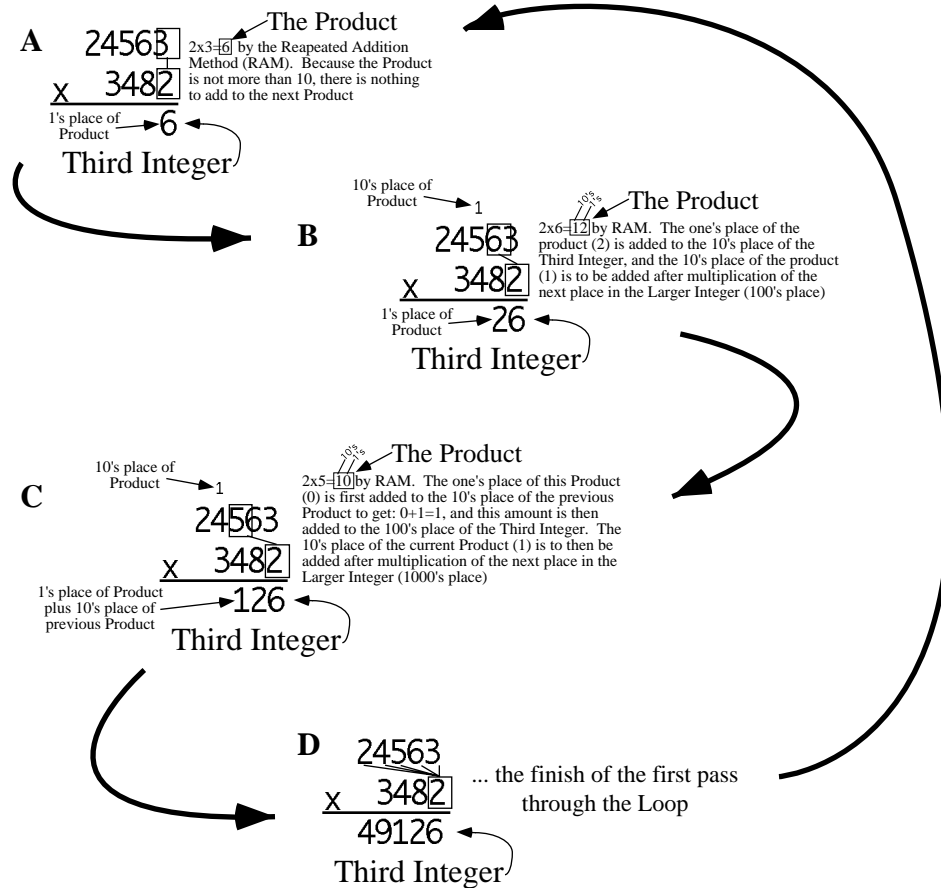


Figure 1.2 - The Loop

Before adding the 0 (1's place of the product) to the 100's place of the Third Integer, however, the carried number of Step B, 1, must be added: $0+1=1$. This amount is then added to the 100's place of the Third Integer. This same procedure would be done with any carried amount from a previous step. As in Step B, there is a 10's place of in the product, so it is then held as the carried number to be added to the 1's place of the next multiplication (which will be of the 1's place of the Smaller Integer and the 100's place of the Larger Integer). Steps A, B and C lay out the basic kinds of instances that can arise when multiplying part of the Smaller Integer with the parts of the Larger Integer: cases where the product less than or more than 10, and cases where the products have carried numbers from steps just prior. The procedures for these instances are then repeated in each step through the places in the Larger Integer until the last place of the Larger Integer is multiplied with the part of the smaller number (Figure 1.2, D). At the last place of the Larger Integer, if there is a carried number, it is simply added to the next greatest place in the Third Integer. This completes one run through the main Loop.

The Loop is then repeated again, except this time using the next place in the Smaller Integer (in our example, it is the 8 in the 10's place) to be multiplied with the parts of the Larger Integer. Everything works the same as above, except the additions to the Third Integer are placed orders of magnitude higher, equal to the place of the part of the Smaller Integer being multiplied. So, e.g., in the second pass through the Loop, since we

are dealing with the 10's place of the Smaller Integer, all additions are now one place higher (the addition being effectively multiplied by 10 — Figure 1.3, 2nd); in the third pass, two places higher (multiplied by 100 — Figure 1.3, 3rd)... etc.

The Loop is run through again and again, until the last place of the Smaller Integer has been multiplied through all the places of the Larger Integer — Figure 1.3 shows the results of the 2nd, 3rd and 4th runs of our example through the Loop. When the final run through the Loop is made, the Third Integer contains the answer, which is given as output. In the figures, I have purposely left the additions to the Third Integer from the constituent runs through the Loop separate so as to distinguish each run's contribution to the Third Integer. (A variation of this might actually create new stores for each run through the loop, which are then added after all runs through the loop are complete.)

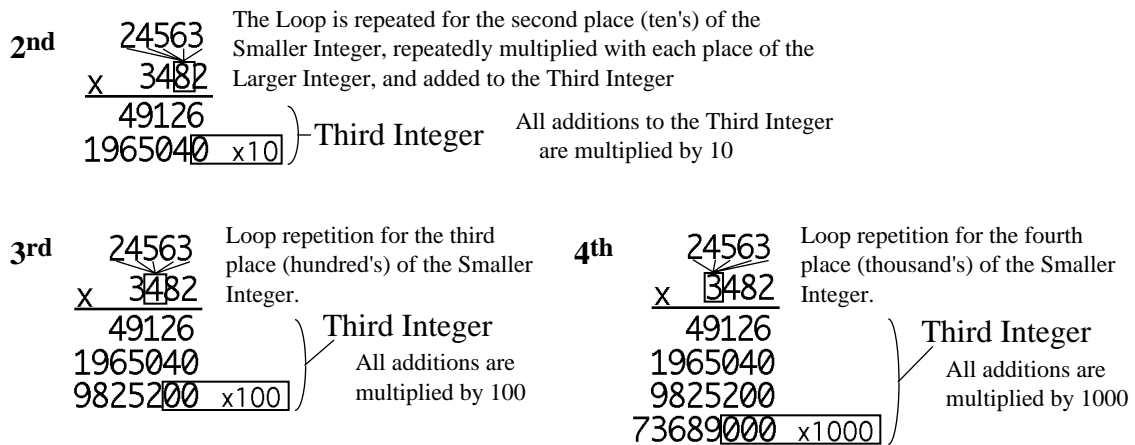


Figure 1.3 - Repetitions Through The Loop

The final output is a summation of these constituent parts:

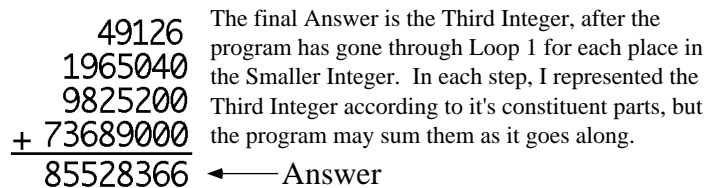


Figure 1.4 - The Answer

This is the general procedure for Program 3. Program 3 is certainly more complicated than Program 2. The main advantage it has over Program 2 is that the difficulties of large integer representation in the RAM algorithm are eliminated in the PPM algorithm by keeping the actual repeated additions to under 10 per multiplied integer, while at the same time still handling multiplication problems of any size.

Of course, the crucial point is that neither Programs 2 or 3 have handcoding in the way Program 1 does. There are, however, two reasons for contrasting Programs 2 & 3. First, merely avoiding a kind of handcoding does not entail having the best explanation: both Programs 2 & 3 appear to avoid handcoding, but we might still argue that Program 3

is closer to how average humans do multiplication. Second (and more important), with the PPM algorithm of Program 3 outlined, we can also consider the following interesting case: Program 3 could make use of a look-up table by replacing the use of the RAM algorithm with a look-up table handling multiplication to two integers from 0-9 each. Is this handcoding? Here it could be argued that the average human multiplier really does memorize a multiplication table handling two multiplied integers up to 9 each — and such memorization might be close to a lookup table. But even here, the general PPM algorithm still remains extensible to any size integers being multiplied; and the answer to the problem does not exist antecedently if either one of the two integers being multiplied is larger than 9. So a look-up table in this case might not be considered handcoding.

This drives home the point that what counts as handcoding depends on your explanatory goals and the nature of what you are trying to explain. In the above cases, we were interested in explaining how average high-school educated humans multiply. So representing memory as a look-up table may not be handcoding.⁷ If, on the other hand, we were interested in how memory really works, having a simple look-up table of any kind would probably not suffice — and if it doesn't suffice, then having such a look-up table would constitute handcoding with respect to explaining memory.

Look-up tables used in the way Program 1 uses them, are one of the most common ways handcoding occurs. There are a variety of different ways look-up tables might be instantiated — in fact, look-up tables are really just one way of talking about matching rules in general. For example, the part of the look-up table in Program 1 which takes inputs 12 and 15 and matches them with 180 is really a rule stating that “if 12 and 15 are input, then output 180.” Matching rules can also come in a variety of forms, including variable kinds in input paired with outputs that may also contain variables. One well-known example of a program using more complicated matching rules is the ELIZA program (Weizenbaum, 1966). ELIZA was designed to simulate a caricature of the behavior of a Rogerian therapist. ELIZA operates by matching the “input” sides of rules against the user's last sentence, and then uses the appropriate “output” side of the rule chosen to generate a response. For example, if I wrote, “My brother is mean to me,” ELIZA might match the rule:

[[{*family-member*} = [brother, sister, mom, dad,...]]]
(My {*family-member*} is *Y* to me) --> (Who else in your family is *Y* to you?)

and subsequently respond, “Who else in your family is mean to you?” The rules in ELIZA are clever and numerous, allowing ELIZA to respond with a roughly intelligent-looking sentence at each turn in the conversation. However, just as Program 1 and its use of a look-up table was found wanting, ELIZA is generally considered to be dissatisfying as a model of how it is believed that intelligent, conversation-capable people work.⁸ In particular, ELIZA gives “canned responses” to questions, as designated by the matching rules, and this is not satisfying for what we mean when we consider the computation that

⁷ Unless it is shown that the process of calculating crucially depends on how memory really works — I will discuss this issue of separability of different facets of a phenomena, and whether they constitute handcoding, in Chapter 2.

⁸ Although some frustrated patients of Rogerian counselling might disagree.

must be performed in order for some “understanding” of the conversation to arise. This is analogous to why we wouldn’t consider Program 1 to really be multiplying as a computation; the analogy would be that the programmer of ELIZA really did all the work in deciding what would be an appropriate response to make it sound like the program was participating in a conversation.⁹

Lurking behind the discussion of the multiplication program examples is another issue which has not yet been addressed, but sheds some important light on identifying what handcoding is. This issue concerns what sets the “correctness” of the task of multiplication itself. That is, in judging the programs, we are looking for them to be capable of multiplying *correctly* — i.e., getting the right answer. There is an important sense in which we might consider whether a human student is “handcoded” by a teacher. And in that sense, then, is learning a kind of handcoding? The math teacher trains students so that they always (at least, that’s the goal) get the correct answer for a given math problem. And the teacher does this by making the student shape her actions so that such actions become (almost) habitual. One might ask what the difference is between a training or learning situation and the sense of handcoding I am trying to develop here.

There are actually a couple of differences. There is a similarity to handcoding in that a trainer (which could generically be thought of as anything that sets exemplars; a “trainer” in this sense could include the environment, both physical and cultural) sets the guidelines (be they causally necessary or normative) for behavior; this is similar to what a programmer does in specifying the rigid causal rules to be followed in a computer program. But in training there is an important difference — the training process is a *two-way* process: there is a “trainer” (whether it is an agent, environment, or culture) and a *learner*. The trainer provides the guidelines which the learner must now work to meet — the learner must still organize itself to meet the criteria of the trainer. Handcoding in the sense that I mean it is instead more of a *one-way* process: the trainer not only sets the guidelines, but also *determines* the activity (including internal activity of the system) that

⁹ This discussion highlights an important side issue: when it comes to computational cognitive science, it *does* matter *how* a program operates — this follows directly from computational cognitive science being aimed at explanation. Alan Turing’s (1950) famous description of the Turing Test is only properly viewed as asserting the valid point for functionalism that intelligence is not a matter of particular appearances or even the physical makeup of the agent (although having a body and some form of appropriate sensors and actuators may be quite important). But, the Turing Test is not sufficient for determining whether something is intelligent *like* a human: it is possible for a program to mimic human intelligence without in fact being intelligent *like a human*, just as our first multiplication program (the look-up table) is able to mimic the function of multiplication for a certain range of input without the capacity to perform the function for even just slightly out of its range (again, brittleness). Behavior alone is not sufficient to determine underlying intensional functionality. It is probably true that in the physical world, for some instantiated function to actually produce the *same* behavior, it would have to be the same intensionally (speaking of intensionally in terms of definition of a function, *not intentional!*), or pretty close — In other words, physics puts constraints on how functions are actualized. But just to be clear, the problem of handcoding does not imply an anti-functionalist claim: something can just as well *not* be handcoded and also not be made of flesh or even look like a human, but still be intelligent like a human. (There are some aspects of appearance and material makeup that do constrain what can be intelligent, but that is independent of the issue of what is handcoded in terms of internal functional organization.) The Turing Test has recently drawn criticism from within the AI and cognitive science community (Hayes & Ford, 1995).

will meet them. Thus, there is really only one process setting behavior, rather than two processes (in which one sets the guidelines and the other works to meet them). This, of course, does not mean that there can't be handcoding in accounts of learning; in fact, in a deep sense, many models of learning to date do not avoid kinds of handcoding in accounting for learning. (I will address this issue in much more detail in Chapter 4.)

(The above should also not be taken to mean that a programmer setting up a program that operates "properly" is committing handcoding in a bad way. Rather, if the programmer is setting the program's operation by a fixed algorithm, it should operate the same way that the phenomenon we wish to explain or reproduce does in nature. Whether the programmer is improperly handcoding a program that is intended to work like some phenomenon in nature ultimately depends on what the ontological status of the phenomenon in nature is. So, just because we build mechanical artifacts, like spring-loaded mousetraps, doesn't mean that we've necessarily handcoded; evolution has *not* "improperly" handcoded certain phenotypic traits into animals; and computational evolutionary models aren't necessarily "improperly" handcoding features of evolving populations. As already suggested, what is proper or improper handcoding depends on how the phenomena being modelled occurs; handcoding is proper if the model works the same way as the phenomena to be modelled works, but improper if the programmer is setting up her program so that it "skips steps" that nature has to take.)

Up to this point, the role of look-up tables, or input/output matching rules in general, have been called into question — they can be likely sources of handcoding. However, as our modification to Program 3 showed, matching rules do not necessarily entail handcoding. Rather, handcoding may be a matter of degree, and handcoding also depends on our explanatory goals. And, when considering phenomena involving training and learning, it is important to consider the nature and source of the criteria for the phenomenon in question, as the relationship between such criteria and the phenomenon itself may constitute a place for handcoding to occur. The simple example of the multiplication programs has already uncovered several important aspects of handcoding. I now turn to present the general concept of handcoding, followed by more detailed examples for clarity.

3 - Handcoding in general

The notion of handcoding in general concerns the relation between: (1) the influence of the programmer, designer, experimenter, user, and/or interpreter — henceforth, I will refer to any possible combination of all five of these as the *researcher* — on the computational model's structure and behavior, and (2) the phenomena that the model is claimed to produce or explain. This makes the identification of handcoding a fundamental issue regarding the explanatory power of computational cognitive models. Handcoding, in its intended derogatory sense, thus means that something has gone *wrong* with the involvement of the researcher in the operation of a cognitive model. Particularly, this involvement is problematic when the "distance" between the problem statement and the solution has somehow been crossed by the researcher "helping the program across," when the program instead should cross that distance itself. In this way, handcoding in a model somehow represents a *failure of the model to meet its explanatory*

goals. Because handcoding is relative to explanatory goals, it should be made clear that the goal in avoiding handcoding is *not* to attempt to remove the researcher entirely from the task of creating the program. Again, there is legitimate researcher involvement and illegitimate researcher involvement. Assuming that the explanatory task of computational cognitive models is to account for the mechanisms that make autonomous, intelligent organisms in fact intelligent, we can then lay out the difference between appropriate and inappropriate researcher involvement (which, in turn, determines whether a model succeeds or fails as an explanation):

- (1) *Appropriate handcoding*: If the researcher codes up a program with structures and processes appropriately similar to those which are possessed by naturally occurring, autonomous, and intelligent organisms, then the program is a legitimate model of those mechanisms; the cognitive scientist, in this case, is no more guilty of handcoding than evolutionary, developmental, and/or learning processes are in evolving actual intelligent species (note: this does *not* entail that the cognitive scientist has to play the same role as evolutionary processes in building models).
- (2) *Inappropriate handcoding*: If, on the other hand, the researcher codes up the program in such a way that the program is still crucially relying on the structures or knowledge naturally occurring *only* in the unexplained mechanisms of the researcher (i.e., these mechanisms are not anywhere in the model, but are only found in the researcher — e.g., in Program 1, the researcher did the multiplying to make the look-up table, not the program itself; the program itself only matched input and output values with no intermediary steps), then the program itself cannot be considered a legitimate model (or at least, cannot be a complete model) of the naturally occurring intelligent phenomena, because such phenomena still essentially exist *unexplained* in the researcher.

The problem of handcoding is actually a special case of the general problems faced by any science which employs theoretical and actual working physical models as part of the explanation of a phenomenon. These problems may be generally distinguished as falling into two complementary categories: the first concerns problems that can arise from *direct researcher influence* on the model, either in the *construction* or *functioning* of the model; and the second kind of problem regards the *interpretation* of the model, and includes interpretation with respect to the *structures in the model* itself, *data produced by the model*, and even draws on potential problems in the *interpretation of the naturally occurring phenomena* that we wish to model. Handcoding likewise comes in these two kinds, and I will subsequently refer to them as handcoding with respect to direct researcher influence on a model and handcoding with respect to interpretation of a model (Figure 1.5). The possibilities of model success or failure just given above are with respect to problems with direct researcher influence on the model. In Section 4 of this chapter, I will primarily discuss this first kind of problem (researcher influence on construction and functioning) in three examples of computational models. I will later address the second kind (interpretation) in Chapter 2, Section 2.3 (under the issue of

measuring theories), Section 5, and it will be a recurring issue in subsequent chapters.

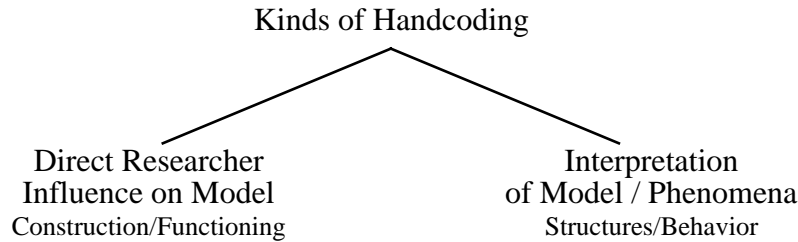


Figure 1.5 - Two Kinds of Handcoding

3.1 - The Analog of Handcoding in Other Sciences

What might be the analog of handcoding in other sciences? As already mentioned, handcoding particularly involves the influence the human researcher has on the operation or interpretation of the model, relative to what is to be explained. In other sciences, problematic influence on model construction and functioning tends to be more obvious. However, handcoding can still occur. I will now briefly discuss what the analogues to handcoding look like in the use of models in other sciences.

3.1.1 - Direct Researcher Influence

The analog to direct researcher influence in models used by some other sciences is best shown in an example. Consider a physics demonstration of how motion of an object traveling through a gravitational field is described by a parabolic arc.¹⁰ Suppose from the outset that we didn't know how objects moved in gravitational fields, so we are depending on this demonstration to somehow give us insight into the nature of this phenomena. An actual physical demonstration of this might include some sort of device that can launch small projectiles of various weights using a variable amount of force, and aimed at different angles towards the horizon. A physicist could then demonstrate, using the apparatus, a variety of different kinds of parabolic arcs traveled by a projectile by actually launching projectiles under a variety of different conditions (different weights, forces and angles). We could measure the "outcomes" of these examples, correlated with certain antecedent conditions, and from these derive general laws of behavior based on the demonstrations made. This situation might then be considered a successful "model" demonstrating at least some features of how gravitational fields affect moving objects (for instance, we might derive a general principle about the ratio of force to mass, distance, and height traveled, given a certain angle). (Of course, the very status of this demonstration as a "model" might be called into question because it is, in an important sense, a literal slice of the naturally occurring phenomena itself; for the purposes of the present discussion, however, this is not as crucial.)

¹⁰ Strictly speaking, the path of an object in a gravitational field is described by some conic section — objects overcome by gravity, as I'm assuming in this physics example, just happen to be parabolic arcs (a kind of conic section). Planetary orbits are circular or ellipsoid, and some comets have hyperbolic trajectories.

It would be pretty obvious something was wrong with the “model,” however, if the physicist always had to demonstrate how objects traveling through a gravitational field moved in an arc by taking the “projectiles” in his own hand and moving them through the air in a parabolic arc. In this case, while it is true that the arcs the physicist makes the objects move in could be appropriate for certain weights, forces and angles, we would be sceptical that his demonstration would be consistent enough for us to draw some general understanding about how the phenomena works (for an answer to “what makes it do that?”). Furthermore, key information is lacking from this “model”: What correlates to “force” in the physicist’s movement of the object so that it could be correlated with particular distances, given angle and mass? (Obviously, if the physicist was merely trying to show us what traveling in a parabolic arc would look like, this demonstration might be sufficient. This again highlights that it depends on what the task of the model is in an explanation; if we’re wanting to learn just what traveling in a parabolic arc is, this would be sufficient, but if we wanted to learn about the invariant relationships which exist between physical properties, such as force, mass, distance, etc., then this is clearly inappropriate — it is this latter goal that I am assuming here).

In taking a model seriously, we want to be convinced that in some way the model *follows the same laws of behavior* or *demonstrates the same kinds of principles* as the system we wish to explain. Our physicist might be very well trained so that he makes the correct arcs in the correct circumstances, but he could also just as easily not move the “projectiles” that way (he could move them in a straight line or a circle, even though the center of gravity is presumably “below” the physics demonstration). Making the judgement that we are to be concerned with the physicist’s involvement in the latter demonstration is pretty simple because we have a good idea of how the phenomena in question should behave — namely, it should work such that given certain circumstances, the projectile *couldn’t* move any other way. It is because of the nature of the phenomenon (that physical laws are consistent under given conditions) that makes this kind of role of a human in the model clearly inappropriate for deriving general principles about movement of objects in a gravitational field.

The reason for this simple contrast is important because it can be compared to the kind of situation we find in computational cognitive modeling. Namely, cognition is exactly what human researchers do — so, in modeling cognition, it becomes a very difficult task to separate (1) the role the researcher’s own cognitive processes play in the setting of the conditions in which the actual processes in the model operate from (2) what the model can be said to “independently” manifest. There is also the additional issue that computational cognitive science is not after merely “simulating” kinds of intelligence — it considers its models to *be* instances of actual cognitive phenomena. The analogous kind of physics model that cognitive science is after is one that actually reproduces instances of the natural phenomena (even if in “scaled-down” versions). Both of these points raise important questions concerning the identity of what intelligence is, and I will address them, below in Chapter 2, Section 2.

3.1.2 - Interpretation

Handcoding of the second type, with respect to the interpretation of a model’s structures or the data the model produces, also occurs in sciences besides computational

cognitive science. This second type was identified and made famous in the philosophy of science by N.R. Hanson (1958) in his book, *Patterns of Discovery*, under the name of “the theory-ladenness of observation.” The general idea of theory-laden observation is that we interpret the information we receive by our senses as being of a certain kind of phenomenon. For example, I take the round, red object in front of me to be an apple. Much of this occurs unconsciously, but our taking it *as* being something is still a product of our underlying theories of what we believe exists in the world, and how those things relate to each other and behave (i.e., a notion of natural kinds arranged in hierarchical relationships).

It is important to pause here and comment on a mistake Hanson’s theory makes — clearing this up will help to refine an important notion I wish to use. Several philosophers of science (notably, Aronson, 1984, p.98; and Harré, 1986, pp.168-172) have pointed out that while Hanson’s intuition was correct, he based it on a mistaken assumption. Hanson took the phenomenon of “seeing something as something else” to be a sort of gestaltic perceptual phenomenon, akin to the “visual gestalt” experiences had with the perception of ambiguous figures. For example, the famous “duck-rabbit,” which can be seen as a duck or a rabbit, or the Necker cube, which can be seen as facing upward or downward. These kinds of figures are interesting in that when we think about seeing one of them one way, we can’t help but see it that way, and *not* the other way (think ‘rabbit’ and see rabbit; think ‘duck’ and see duck; but you can’t see both at the same time; try it with Figure 1.6) — yet clearly the figure on the page we are looking at isn’t literally changing as we see it as being different things.

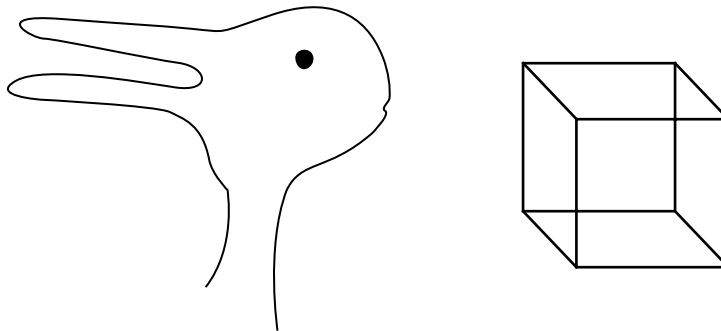


Figure 1.6 - The “Duck-Rabbit” and Necker Cube

Hanson was drawing an analogy based on the following difference: that the difference between *what* it is we are looking at and what something is “seen as...” when we apply particular beliefs or concepts (like “this is a duck” or “this is a rabbit”) is like the difference between the raw information in the world and how we then interpret and organize it according to some conceptual ontology.

The analogy, however, breaks down when considering actual cases of scientists making judgements based on observations in support of or against a theory. Seeing something *as* something is what is going on in the visual gestalt situation. However, making a judgement as to *what kind of phenomena* a particular observation is *of* (which is what scientific observations that are taken as evidence, data, or predictions are doing) is a case of “seeing that...” In this case, there is also an application of concepts into making sense out of the observation, but there isn’t necessarily a gestalt experience of the kind

like the visual gestalts.

Harré uses the example of a comparison between what Lister and Pasteur saw when looking through their microscopes at infected wounds. Lister thought micro-organisms swarming around the damaged flesh were human cells which had “gone wild,” while Pasteur thought they were hostile invaders. If Lister changed his mind to agree with Pasteur, there’s no reason to believe that any difference occurs in what he sees with the help of his microscope. Rather, he would fit such an observation into a different conceptual framework — rather than taking his observation as of wild human cells, they would be of hostile invaders; but the perception would be the same — unlike the perceptual gestalts in which the perception does change (even though the actual external object does not). Again, the idea of theory-ladenness as the tendency to interpret a situation in a particular way according to our background theories does still hold, so long as we are careful not to require it to be a kind of visual gestaltic-perceptual experience.

The potential problem that theory-ladenness presents is that we, as scientists, might come to interpret particular observations we make as being a kind of phenomena, when in fact a better explanation would take them to be something different. Even observers who are alerted to the fact that they might be misinterpreting cannot easily escape their own particular theory-ladenness of observation (in an important sense, it is impossible to make any sense of the world without some background theories to organize it into something meaningful — an intuition which Kant had). This phenomenon has been well documented in psychological studies which have shown that people’s background assumptions can greatly influence the way they interpret the world (for example, “priming” effects on people’s decision-making are well documented: Meyer & Schvaneveldt, 1971; Schacter, 1987; Tversky & Kahneman, 1974; see also Bruner, 1957).

In most sciences, problems with handcoding with respect to interpretation occur in the interpretation of data in support of their theory. There have undoubtedly been numerous cases where scientists convinced of a theory have come to see confirming evidence of their theory regardless of the actual data presented. At other times, the scientist may even try very hard to not over-interpret or bias their judgement, yet must still end up making some judgement about an observation that isn’t “directly” made evident in the observation itself.¹¹ The relevant point here is that observations, in order to be taken as being of a certain *kind* of entity or event, must be interpreted as such; observations themselves do not automatically provide their own interpretation. Such interpretation is made in an additional assertion, which is called a *measuring theory* — I will discuss these kinds of theories and their relations to models in more detail in Chapter 2, Section 2.1.3.

The point I wish to highlight here is that making theory-laden interpretations of the structure of the *model* itself, while seeming more rare than in cases of interpreting what a phenomenon nature is, is also possible — and this is my chief concern with handcoding interpretation (in Chapter 2 I will distinguish theoretical models from physical model

¹¹ Several examples of cases such as these occurring in the history of early sub-atomic physics are given by Hanson in another book of his, *The Concept of the Positron* (1963). In these cases, observations were made which could easily confirm several rival hypotheses, and particular interpretations were made which took the observations to be supporting a particular hypothesis.

exemplars — the latter is what is subject to our theory-laden observation, even if it is intended to be a “physical instantiation” of our theoretical model). Sciences that are relatively young will tend to have a greater potential for this problem because there is less agreement on what counts as an instance of a kind of phenomenon. The most prominent example in cognitive science is in regards to what is taken to be representational, or count as having “meaning about something else”: it is easy for us to see something as meaningful because we are very good at attributing meaning to events and objects; but whether something has meaning in and of itself about something else, independent of what we might attribute to it, is a very contentious issue. Coming up with a theory of representation that removes the possibility of accidental semantical attribution is therefore a very difficult issue.¹² This issue will be central in the coming chapters.

Such attributions of the status of certain *kinds* (even to actual physically working models) entail further assertions (measuring theories), and this highlights a very important point: models do not in and of themselves represent “existence proofs” of how some phenomenon is produced — they only constitute a possible example of some phenomenon when accompanied by a theory of how such a phenomenon would be manifested in certain circumstance. And they fail as such an example if it is found that such a theory of how the phenomenon would be manifested is shown to be false.

This point can be easily misunderstood when considering a working computational model. The tendency is to create a model that can produce some desired behavior, and then argue that because such behavior was produced, the model must somehow constitute an explanation of the phenomena in question. This demonstration does not in itself constitute an explanation; it must be accompanied by confirmation that the circumstances surrounding the production of the behavior in the model are consistent with the kinds of circumstances that the naturally occurring phenomena to be explained likewise occur in, and that the natural kinds in the model correspond (within an agreeable degree of accuracy) to the natural kinds in the system to be explained. A system may be able to perform a task, but in terms of an explanation of, say, how a *human* does something, it’s a theoretical question as to whether the model does it *like* a human (or other animal). While a model may be able to perform a task, it may really only have manifested its capacity because it is performing under limited conditions which are far too simple compared to what actual cognitive agents have to operate within.

So, for example, just because a computer program is capable of parsing a page of English text, that performance alone does not entail that the computer program is capable of “reading” in the same sense that literate humans can. There are many other criteria that must be met: what constitutes having “read” or “understood”? How does the computer model do the “reading”? Are the processes in the computer in any way related to the proposed processes in the human reader’s brain? In the background of all of these questions reside explicit or implicit acceptance of particular measuring theories (which pertain to the *identification* of kinds of entities and processes) in addition to theories concerning how the processes might work.

¹² This, of course, assumes that there is a notion of representation which constitutes a natural kind — a kind of state or process of nature which is “naturally” representational independent of observers who attribute some sort of representational relationship.

Examples like this also show how handcoding with respect to interpretation and to model construction and function often go hand-in-hand (their deep relation will be exposed in Chapter 2). If there exists a basic misunderstanding about what the phenomenon to be explained is, then both types of handcoding can occur: (1) the researcher, in constructing the model, may have given the model inappropriate conditions for manifesting the phenomenon, or even interfered with the actual model operation thus making the model appear to manifest a certain behavior which ultimately only the researcher is capable of; while at the same time (2) the researcher will be improperly interpreting particular model behavior as instances of that phenomenon. Nonetheless, it is important to keep these two kinds distinct for the purposes of analysis.

3.1.3 - Why is Handcoding So Non-Obvious for Cognitive Science?

The above example of the physicist moving the object in a parabolic arc — of how a physics model might be handcoded — is very unsophisticated; in fact, it is downright trivial. More subtle handcoding in physics as well as other scientific models can take place and has. One of the famous cases in early study of animal intelligence is that of “Clever Hans,”¹³ the horse which was reported to “add” numbers which its trainer verbally commanded it to add: the trainer would say, e.g., “Hans, add 5 plus 7,” and Hans would answer by tapping his foot to the total added amount. After several demonstrations, the horse did appear to manifest the amazing ability to add if given a vocal command to add two numbers. Unfortunately, it turned out that Hans did not have quite the ability for which he was originally billed; Hans was instead extremely sensitive to the facial expressions and body-position of his trainer (and others present), so that as the trainer was counting off the numbers in *his* head, his facial and body language changed subtly to mark nearing the end of the counting, at which time Hans stopped tapping — Hans wasn’t exhibiting the ability to add, but the ability to read body-language; Hans’s tappings also exhibited his *trainer’s* (or others present) ability to add. It is not clear whether the trainer in fact knew of his subtle cuing. Regardless of whether it was intentional or not, it was a clear case of researcher (or, in this case, trainer) interference in the production of the phenomena — removing the trainer and anyone else present from Hans’s sight resulted in Hans’s loss of the ability to add.

While the transparency of such examples makes handcoding with respect to construction and functioning of most scientific models seem to be rather “obvious” (and this may often be the case for many sciences), this issue is not as obvious in computational cognitive science. This is particularly clear in the situation concerning interpretation of models and their behavior. Why is the issue of handcoding so sticky for computational cognitive science? — Why isn’t handcoding in computational models as obvious as the above example of the physicist moving an object through the air as a *model* of flight of a projectile through a gravitational field? I believe there are two reasons for this.

¹³ This example almost bears the status of folk-lore, but contains a good moral to be reminded of in ethological study and for identifying handcoding: knowing as much as possible about the ecological niche of the animal being studied is crucial to understanding that animal’s behavior.

First, computational cognitive science is still relatively new: being generous, the science has been around for only about 40 years (beginning with the emergence of research combining psychology and AI). The criteria that develop in particular branches of science for designating what counts as a good model as opposed to a failure in explanation emerge only gradually — periods of theoretical and methodological development are found in the beginnings of any science, and this is recognized in the general history of the philosophy of science. There have to first be *some* models and attempts at explanation, or at least sketches of what such models and explanations might look like (in general, there's no substitute for the genuine article) for there to be some grounds for discussion of proper vs. improper modeling, or adequate vs. inadequate explanations. Computational cognitive science's close cousin of experimental psychology is just a little bit older, and there are quite rigorous standards for proper empirical investigation. But computational cognitive science is still in the process of gaining enough experience as a science to establish such criteria of correctness for model building. Computational modeling, in particular, now needs the same kind of rigorous methodological foundations to make it as empirically respectable as other sciences, such as biology, or experimental cognitive science. It is my desire to have this chapter make some headway in finding such foundations for computational cognitive modeling.¹⁴

The second reason handcoding is such a slippery issue is that computational cognitive science is attempting to explain what amounts to being precisely the *same* phenomena that are produced by cognitive scientists themselves: intelligent behavior / cognitive capacity. It can therefore be very difficult to separate the researcher's own mechanisms for the production of intelligent phenomena from the operation of the mechanisms in the model itself, especially since the model itself is a product of the researcher. (As already discussed above, it also makes proper interpretation of a model's accomplishments very difficult; theory-ladenness of observation makes much of the desired goal of "objectivity" in computational cognitive modeling very difficult to achieve.) Nonetheless, just as with any other science employing working physical models, it should be possible for computational cognitive science to also produce working computational models which do exhibit and explain intelligent phenomena independently of any reliance on the cognitive

¹⁴ It is also arguable that the introduction of the computer, a vastly flexible tool that can be used to "envision" possible descriptions of the world, has likely introduced so much flexibility that issues of the interpretation of models themselves comes clearly to the foreground. This does not invalidate computational models as scientific models (as some mistakenly argue; Hendriks-Jansen, 1996); instead we need to be more explicit about what in the computational model is intended as an instance of what kind. I will discuss this in more detail in Chapter 2.

mechanisms of the cognitive scientists themselves that created them.¹⁵

The issues and instances of handcoding in cognitive science and AI have not gone unnoticed. In fact, handcoding has been identified by many researchers in the history of AI and cognitive science who have taken a critical look at existing models. These identifications and criticisms have not just spelled some failure of a particular research program. Rather, these recognitions have more importantly become the hallmark of great advances and learning with respect to the understanding of the phenomena under investigation, as well as the development and evolution of the methodology of AI and cognitive science. And while focus has often been on the negative aspects of handcoding, I believe it is important to also highlight its positive contributions. In fact, debates over handcoding issues have created so much intellectual ferment that they can be taken as one of the great achievements of the computational paradigm: the power of computational cognitive modeling is that it forces the researcher to make explicit the ontological commitments made in the theory(-ies) funding the explanation which aren't directly found in the discourse describing the model. Computational models also make their explanation publicly observable in a way not possible through discourse only. This methodology fleshes-out strictly theoretical assertions and brings them into an empirical forum, while also making explicit what the models do *not* account for.

Several modeling techniques which have become subfields of AI are explicitly devoted to removing the amount of handcoding possible in the creation and operation of computational models — most notably: evolutionary computation, which includes genetic algorithms and genetic programming, as well as a variety of other kinds of stochastic search methods (Angeline, 1993; Goldberg, 1989; Holland, 1992; Koza, 1992).¹⁶ Connectionism also naturally addresses some aspects of handcoding, e.g. through gradient descent learning, in which the network “settles” on its “own” representation (input-output pairing function) — I will discuss this more, below.

¹⁵ One might argue that this is true except in the basic task of recognizing something as intelligent in the first place. While it is true that our general common-sense definitions come from normative judgements about what is intelligent and what is not, cognitive science generally seeks a notion of intelligence that is naturalized. For example, one possible naturalized notion of intelligence might be: “*intelligence* is the having the capacity to go beyond the information given” (Bruner, 1957). This capacity could then be considered as a natural kind of complex, epistemic system organization. Another possible naturalizable notion of intelligence would be “*intelligence* as adaptive behavior” (Beer, 1990). Regardless of which particular working definition is adopted, it should also do justice to our intuitions as well (i.e., it should not be completely arbitrary). Additionally, we can still point to a particular kind of system behavior (independent of whether it's indeed “intelligent”) and ask how it works.

¹⁶ I should note here that these methodologies are not immune to handcoding; often very extensive pre-computational analysis on the task domain is done prior to the use of evolutionary computation methods, and this analysis comes primarily from the intuitions of the human researchers. As will become clear, this does not necessarily entail that inappropriate handcoding has occurred, but does present the possibility. Thanks to Bram van Heuveln for pointing this out to me.

4 - Three Handcoding Case-Studies

I will now present three computational models which have been previously recognized by researchers in AI and cognitive science as involving problematic handcoding. These three examples have been chosen because they do represent fairly straight-forward versions of handcoding and introduce several of the ways in which models can fall prey to handcoding of the first kind: inappropriate involvement of the researcher in the *construction* or *functioning* of the model.

4.1 - AM & EURISKO

One of the most well-documented cases of inappropriate handcoding in the history of AI is the AM program. An extensive and detailed analysis of the AM project is found in Ritchie & Hanna (1990), as well as Lenat & Brown (1984).¹⁷ AM was created by Douglas Lenat as his dissertation (1976, 1979, 1982, 1983). AM is an acronym for “Automatic Mathematician.” The program was intended to be a model of machine learning of concepts through a constructive search process based on simple rules and a set of axioms (base-primitives). EURISKO was a later improvement on the basic AM architecture. Among its improvements was a “learning” mechanism to construct heuristic rules while engaged in search.

AM’s operation was specified by taking “knowledge structures” of represented mathematical axioms and then a set of rules for performing inferences based on those axioms. The starting axiom set included about 100 basic mathematical concepts amounting to set-theoretic fundamentals and the notion of cardinality.

Also included were about 230 heuristic rules (“rules of thumb”) which rated what aspects of represented mathematical structures are important or “interesting.” These heuristics were based on what Lenat took to be the kinds of structures that mathematicians generally find interesting. Some examples of these heuristics are: whether the new structure included multiple instances of simpler operations, such as a square being a number multiplied by itself; or the inverse of an interesting concept, such as a square root is an inverse of a square; or how “unique” the new structure is. Structures were rated by a numerical value according to “how interesting” they were.

Given these base specifications, AM would then search for new and interesting mathematical structures. The search would involve constructing new theorems based on the original axioms, or derived theorems, according to the set of rules provided. The generative process of creating new structures was not exhaustive, but was directed toward interesting areas of the vast space of possible structures: if the structures created were “interesting,” according to the heuristics given, then AM would pursue further construction based on the “interesting” structure made; if the structure was not as interesting, AM would try other paths of search.

It was claimed that AM was capable of making interesting mathematical discoveries based on the heuristics and rules provided. AM was claimed to have discovered, for example, “prime numbers” because it constructed and rated as highly interesting the

¹⁷ Koza (1992, pp.232-236) also gives a good discussion, with particular attention towards distinguishing genetic searches from AM’s kind of operation.

identification of numbers which have a minimal number of factors. Following this discovery, AM went on to formulate the Golbach conjecture, that every even number greater than 2 is the sum of two prime numbers.

The problem is that AM turned out to not do what it was originally thought to do. Handcoding occurred in two ways:

(1) The first problem was that several of AM's key discoveries may have relied on the fact that the mathematical axioms were represented in LISP. As Koza (1992, p.232) points out, many of the mathematical concepts and heuristics in AM were stated directly in terms of lists. Lists are the primitive data type of the LISP programming language. An example of this is that in some of the mathematical structure representations, integers were represented as lists of *t*'s, where *t* denotes "true." Thus, the integer 5 was represented as the list (*t t t t t*). Furthermore, the lists in AM were manipulated by functions that are native to LISP. For example, the LISP function APPEND concatenates two lists, CAR returns the first element of a list, and CDR returns the tail of a list. So, when an integer such as 5 is represented (*t t t t t*), the LISP list-manipulation function CDR has the effect of subtracting 1. Likewise, when two integers are to be added, the LISP list-manipulation function APPEND has the effect of adding the two integers.

In Lenat & Brown's (1984) paper, "Why AM and EURISKO appear to work," Lenat admitted that AM's discovery of various mathematical concepts was greatly facilitated because AM's concepts and heuristic rules were stated in terms of LISP's primitive "list" object, and these concepts were then manipulated according to list-manipulation functions native to LISP. Put in terms of search through a space of possible generative combinations, the "interesting" mathematical concepts were "denser" (meaning that they were more easily found) in the LISP space than they might be in some other unspecified space (Koza, 1992, p.235).

LISP can be considered to be an antecedent human discovery of a format to represent mathematical concepts and operations in a systematic way to express computational algorithms. Having the operation of AM's search rely on the structures of LISP *for the search* means that the previously discovered mathematical concepts in LISP should also be counted among AM's starting axioms and operations. This is considerably more than knowledge of sets and cardinality. (And thus, AM's search was more of an expansion on the relation between LISP and mathematical structures than on the discovery of novel mathematical concepts.)

(2) The second problem was raised in Ritchie & Hanna's (1990) paper, *AM: A Case Study in AI Methodology*. They highlighted the point that Lenat decided, *during* AM's search, whether a particular line of investigation was "interesting" or not. Thus there was direct human intervention in AM's search. This led Ritchie and Hanna to conclude that personal intervention may have contributed more to the reported results than the automated process.

As a result of these two problems, it seems as though AM's "discoveries" by and large came from humans, rather than from the program itself. In fact, as Hofstadter notes, "This suggests that it might be more appropriate to think of AM as having been a human-machine hybrid rather than as an autonomous computer program" (1995, p.476).

In summary, two kinds of handcoding were committed in the creation and operation of AM:

- (1) Influence of structures inherent in the program which were previously not recognized as lending to the program's achievements (the first problem);
- (2) Active guidance of model behavior by the researcher toward the desired behavior, when it was claimed to do so on its own (the second problem).

Thus, there was clearly *ad hoc* human involvement in crucial steps of mechanism (both by the researcher during the program run, and in implicit structures and operations of the LISP programming language, in which the original axioms were represented), and AM fails as a general explanation or model of autonomous machine mathematical investigation or discovery (at least, to the extent originally billed) because of this handcoding.

4.2 - BACON

Another case-study in handcoding issues is the BACON program (named after the British empiricist, sir Francis Bacon), created by Pat Langley *et al.* (1987). BACON consists of a family of heuristic techniques for inducing scientific laws from empirical data. The creators claimed that BACON has made a number of scientific discoveries, including: Galileo's law of uniform acceleration, Boyle's law of ideal gases, Ohm's law of electrical resistance, and Kepler's third law of planetary motion.

One of the most extensive criticisms of the BACON project comes from D. Chalmers, R. French, and D. Hofstadter (1992; I will refer to all three authors as Chalmers *et al.* hereafter), in their paper, *High-level perception, representation, and analogy: A critique of artificial intelligence methodology*. In particular, they are dismayed at the strong claims made by Langley *et al.*, that BACON is "an accurate model of scientific discovery." Chalmers *et al.* note that it is claimed that Langley *et al.*'s system is "capable of representing information at multiple levels of description, which enables it to discover complex laws involving many terms." Furthermore, one of the creators of BACON, Herbert Simon, claimed that BACON can accomplish these discoveries, "starting with essentially the same initial conditions as the human discoverers" (Simon, 1989, p.375).

Making such "re-discoveries" is a particularly strong claim because the cognitive processes that must have gone into producing such laws were certainly complex. Galileo, Boyle, Ohm and Kepler were brilliant men, and it took several years each for their work to come to fruition. If BACON is somehow able to make such discoveries, and in little time, then it is certainly a breakthrough model of scientific discovery.

BACON starts with a set of values of an independent variable and the associated values of the dependent variable and produces a mathematical expression relating the dependent variable to the independent variable (Koza, 1992, p.255).¹⁸ An example of one of the more complicated laws that BACON is reported to have rediscovered is

¹⁸ Koza, 1992, pp.255-257, gives a good summary of BACON's operation — again, particularly in comparison and contrast with genetic search methods.

Kepler's Third Law of Planetary Motion, which states that the cube of a planet's distance from the sun is proportional to the square of its period. Expressed in mathematical terms (D =planet's distance, p =planet's period, c =constant):

$$\frac{D^3}{p^2} = c$$

There are three stages that BACON requires of the researcher to set up for a search for an answer. BACON first requires the researcher to identify the set of independent variables that are to be used to explain the relationship between the independent and dependent variables. Next, BACON requires the researcher to supply a set of heuristics that might be used to express the unknown relationship between the independent and dependent variables. Koza (1992, p.256) gives two examples of these heuristics:

- (a) If the values of one numerical variable increase while those of another variable decrease, then consider multiplication to be the explanation.
- (b) If the values of two numerical variables increase together, then consider division to be the explanation.

And third, BACON requires the researcher to select a sampling of pairs of values for a representative sampling of combinations of the independent and dependent variables. BACON applies an error measure to the differences between the values of the dependent variable produced by BACON and the values of the dependent variable associated with each step of a search for a solution.

BACON then operates as follows: It first tests whether any of the researcher supplied heuristics are applicable to the given set of sampling data. If a heuristic is applicable, then BACON considers that the two variables are related via the function identified by the heuristic. It then adjusts the sampling of data using the function identified by the heuristic so as to create an adjusted sample set. BACON then retests whether any of the researcher-supplied heuristics are applicable to the adjusted sampling. (For example, if the condition of heuristic (a), above, is applicable to the given data because increases in one variable co-occur with decreases in the other, then BACON considers that the two variables are related via the function of multiplication.) Next, BACON adjusts the given sample data by applying the function that was identified. The adjusted version of the sample data is tested anew to see if any heuristic is applicable.

In this way, relationships involving multiple applications of the functions in BACON's function set (the functions outlined by the heuristics) can be "discovered." That is, BACON discovers a composition of functions from its available function set. If and when the adjusted sampling of data produces an identity (empirical data are usually used, so perfect identity is not a necessary requirement; a margin of error is then allowed — but it must be preset by the researcher) between the independent and dependent variable and the adjusted version of the dependent variable, BACON's search is considered successful and the run terminates. At a completed run, BACON has produced a sequence of applications of functions (i.e., a composition of functions) relating the independent variable and the dependent variable.

What bothers Chalmers *et al.* so much about the program's operation and the subsequent claims made, based on its performance, is that in Langley *et al.*'s set-up of the problem for BACON, all the hard work was done. In other words, BACON is given *precisely the data required to derive the law*, so that its "discovery" is reduced to a rather simple deduction that any beginning physics student should be able to make. For this reason, Chalmers *et al.* accuse BACON of having *20-20 hindsight* in being given only what data is needed, and the data is in just the right form so that the deduction becomes relatively trivial. In fact, the actual cognitive processes involved in such scientific discoveries are faced with the much more monumental task of paring down what is relevant and making careful hypotheses paired with testing, all part of an intricate and complex process of induction.

And this is no small matter. In considering the kinds of cognitive representations involved, we are not talking just about how equations are represented, but whole background metaphysical models based on historical, religious and cultural-framework assumptions. As Chalmers *et al.* point out, this whole background context, and the need to break away from it, seriously complicates the kind of creative processes required by many orders of magnitude; quite a bit beyond the "trial-and-error based on heuristics" type search that BACON used.

"Within [Kepler's historical] context, it is hardly surprising that it took Kepler thirteen years to realize that conic sections and not Platonic solids, that algebra and not geometry, that ellipses and not Aristotelian "perfect" circles, that the planets' distances from the sun and not the polyhedra in which they fit, were the *relevant* factors in unlocking the regularities of planetary motion. In making his discoveries, Kepler had to reject a host of conceptual frameworks that might, for all he knew, have applied to planetary motion, such as religious symbolism, superstition, Christian cosmology, and teleology. In order to discover his laws, he had to make all these creative leaps"

(Chalmers *et al.*, 1992, p.192).

Chalmers *et al.* point out that Langley *et al.* account for the large amount of time Kepler needed — 13 years — to make his discovery as "sleeping time, and time for ordinary daily chores, and other factors such as the time taken in setting up experiments, and the slow hardware of the human nervous system (!)" p.193. Furthermore, Qin & Simon (1990) conducted an empirical study which found that starting with the data that BACON was given, university students could make essentially the same 'discoveries' within an hour-long experiment; The authors, however, took this as evidence in support of BACON, rather than taking this as clear evidence that something is wrong with BACON's methodology.

Chalmers *et al.* conclude that BACON is thus guilty of a kind of handcoding in which the program is claimed to be capable of accomplishing some task requiring cognition, but the really difficult work of perceiving the problem has been bypassed. What was the true feat of Kepler, his real discovery, was the novel conceptual framework he constructed (paramount to a *paradigm* shift, in Kuhnian — Kuhn, 1970 — terms) in which the particular Third Law of Planetary Motion is framed. In BACON's case, the really

interesting and complicated cognitive task of construction of the representation of the problem is bypassed, existing really only in the historical records of Kepler's work and what we have since learned from them — BACON's 20-20 hindsight. The interesting cognitive mechanism that remains to be explained is how such a framework was represented, and how that representation was constructed.

BACON's handcoding is similar to AM's in that the researchers provided the program with far more information than was reported, but highlights how this information was given in terms of tailoring the representation of the problem domain to such an extent that the achievements of the program are relatively trivial compared to the phenomena to be explained. Also, Chalmers *et al.* have made clear the results such researcher influence has on the explanatory power of the model:

- (1) The problem domain that the program is intended to explain is represented in such a way that the cognitive mechanism of interest is bypassed: the program has 20-20 hindsight; not at all faced with the kind of problem faced by those whose processes the program was intended to explain.

In other words, the program doesn't have to go through the computational task of picking out what is relevant from what is not relevant, representing these relevancies in a form that is usable, and then finding a solution. This task constitutes the meat of scientific discovery, and leaving it out is to leave out the very phenomena that BACON was intended to explain.

4.3 - A Connectionist Past-tense Learner:

Rumelhart & McClellon's past-tense learning network

The modeling paradigms of connectionism and parallel distributed processing (PDP) offer some natural answers to the handcoding problem. In particular, connectionism is comprised of a class of learning techniques (e.g., gradient descent learning) in which the artificial neural network "settles" into its "own representation" of some cognitive domain. The advantage that connectionism gains here is that these learning techniques do not require the researcher to explicitly designate the representational units or what they correspond to. Instead, how the network represents is a function of its exposure to a training set of input/output pairings over a series of training "epochs," in which the weights of the network are adjusted to best satisfy the desired input/output pairings via a training algorithm. Here, the influence of the researcher in the designation of the form of representation in the network is minimized — thus decreasing the potential for malevolent handcoding in certain explanatory tasks.

This feature of connectionism is certainly an advance for capturing many kinds of phenomena; but the spectre of handcoding can still visit connectionism if care isn't taken. One of the ways connectionist approaches can fall prey to handcoding is in how the training set is chosen and presented to the connectionist model.

The training set of input/output pairings used to train a connectionist system is crucial for how the net trains ("settles"). In particular, the order in which information is presented can have a profound effect on how the network trains. In learning systems, this is crucial; if the order used in a model does not naturally occur in the world, and thus

requires a researcher to impose such order, then the researcher is handcoding potential phenomena that might not otherwise naturally arise in a model ordered following more natural conditions. Thus, such models which require the researcher to specify the order of the training set in order to induce the phenomena are not good explanations of the natural phenomena — the unexplained ability of the researcher to know what's appropriate (knowledge not available to the naturally occurring mechanism) is being snuck into the successful performance of the model.

Clark (1993), p.155-156, presents an example of this kind of handcoding, using Rumelhart & McClelland's (1986) connectionist model, which is intended as connectionist account of the learning of past-tense verbs. The model was claimed to reproduce the well-documented "U-curve" in performance during learning: in normal child language development, there is a phenomenon in which regular and irregular verbs are initially distinctly treated so that initial performance is good with a limited set of regular and irregular verbs; but then performance degrades through misapplication of past-tense regular verb rules to irregular verbs until, finally, performance again improves with the eventual distinction between proper past-tense rule application to regular verbs, distinct from irregular cases.

This phenomena was initially explained by a model in which there is development of distinct computational mechanisms in the language-tense learner: one for rule-learning, and one for rote-memorization of irregular tenses. Rumelhart & McClelland's model was likewise able to reproduce the observed developmental data, but did so with essentially one computational architecture (that of the network) — change only occurred in weight changes during training. This connectionist model was thus an interesting challenge to the initial multi-mechanism model.

It turned out, however, that while the Rumelhart & McClelland-model's computational structures were essentially the same, the network behavior during training was a direct result of the specific temporal changes in the training set provided by Rumelhart & McClelland. That is, the model's reproduction of the U-curve data was a direct effect of the statistical transition, during training, between a stage in which a high proportion of the data consisted of irregular verbs to a subsequent stage in which the majority of the data consisted of regulars.

As Clark describes the position of the critics of Rumelhart & McClelland's model (particularly, Pinker & Prince, 1988): "... this is not ... psychologically interesting, since human over-regularization errors occur without the benefit of such convenient manipulations of the input data, and reflect not the changing statistics of a training set but the attempt to impose rule-involving order on a body of stored knowledge." Thus, Rumelhart & McClelland's model is in a situation similar to BACON's: there is 20-20 hindsight — the problem was presented to the model in "just the right way." This introduces new issue: not just system operation, but also its task environment can hide subtle researcher influence, amounting to handcoding.

So, handcoding can even take place in connectionism: Rumelhart & McClelland have essentially provided what is the key to reproducing the U-curve during the model's development by changing the training set themselves to induce the U-curve change, rather than a developmental mechanism in the system itself in an environment of homogenous stimuli variability.

The point of this example is not to argue that Rumelhart & McClelland's model does not pose an interesting challenge to the traditional model, nor to argue that connectionism in principle couldn't find ways of explaining language acquisition. In fact, Clark (1993, pp.156-170) goes on to describe the interesting work by Plunkett & Marchman (1991) in reassessing the U-curve data itself, from a decidedly connectionist perspective. Rather, Rumelhart & McClelland's model highlights the potential for handcoding which exists in connectionist methodology (e.g., in the kind of training data and/or the temporal order in which the training data is presented to the network) — i.e., there exists the potential for the success of a connectionist model at producing some behavior to be a matter of unexplained influence of the researcher.¹⁹

5 - Summary

With this introduction to the handcoding problem in general, and some of the instances in AI where handcoding has been identified, it should be clear that there are several kinds of handcoding. With AM we saw handcoding in two ways: the active guidance of model behavior by the researcher, and the influence of programming structures inherent in the program that were previously not recognized as lending to the program's achievements. BACON was shown by Chalmers *et al.* to have been set up in such a way that the problem domain that the program is intended to explain is represented so that the cognitive mechanism of interest is bypassed, thus providing an unfair version of "20-20 hindsight." And even connectionist models were shown to be vulnerable to handcoding, as evidenced in Rumelhart & McClelland's model. Here the situation is similar to BACON's in that 20-20 hindsight was afforded in the problem being presented to the model in "just the right way." This, however, uncovered that the scope of handcoding is not limited to just the internal operation of the system — even the task environment in which the system operates might hide subtle handcoding. If the task environment presented to the model is not representative of the kind of environment that the naturally occurring systems that we wish to explain have to contend with, then the model is not having to solve the same kinds of problems, and therefore the model does not explain how naturally occurring systems contend with those problems. Other kinds of handcoding may be found in the literature, and there are sure to be more discovered in the future.

A rigorous definition of handcoding cannot be found that would unilaterally identify all the specific instances of handcoding. Also, what is observed to *not* be handcoded today may be discovered to be so in the future. In an important sense, any model will involve handcoding of some sort — that is what it *is* to *be* a model. This does not mean, however, that identifications of problems with handcoding are in some way vacuous. Rather, this is a direct result of the fact that the identification of, and attempts to avoid,

¹⁹ Certainly, an obvious place for handcoding issues is in *supervised learning* in PDP systems. Learning in these cases requires that the desired output already be known in order to generate what changes must be made internal to the system to get it to match the desired output. While such learning may naturally occur, much learning is distinctly unsupervised: it is up to the learner to determine what internal changes are required — i.e., to determine what *is* optimal output in the first place.

particular kinds of handcoding are integrally dependent on the current competing theories funding the particular models, and the positions that are reacting against those models (the claims made above that these models involved handcoding, for example, all themselves rest of background theories). The main question before us is then: In what sense is handcoding legitimate, and when is it illegitimate?

Recall that this question was raised back in the beginning of Section 3, and two descriptions were given which distinguished proper from improper involvement of a researcher in a model (either in creation and operation, or in interpretation of the model). In the space since then, I have presented several examples of how handcoding can be illegitimate, and what forms that illegitimate handcoding may take. The question of the legitimacy of handcoding, however, still requires development so that it can be used to diagnose other projects. This development will entail making clear what I mean by computational cognitive modeling, the research paradigm for which the framework for the identification of handcoding is to be made a useful methodological tool. I turn now to Chapter 2 to define computational cognitive modeling and subsequently show how existing handcoding may be identified in computational models. The result is a framework for establishing a motivation for new cognitive theories and better computational models.

Chapter 2

Handcoding and Computational Cognitive Modeling

1 - Introduction

As explained at the end of Chapter 1, the goal of this chapter is to fill out the framework for the handcoding critique with respect to computational cognitive modeling. This framework will enable me to use the handcoding criticism as a critical tool for investigating the legitimacy of computational models as explanations, and provide the guidelines for proposing criteria for future theoretical models which eliminate previous handcoding.

The first major task is to define the research paradigm for which the handcoding critique is to be used: computational cognitive modeling. This will be the property of Sections 2 through 4, which will make up the bulk of this Chapter. Once this definition has been given and discussed, I will present a concise statement of what handcoding is with respect to computational cognitive modeling, followed by discussion of several additional properties of handcoding. This will complete the general discussion of handcoding begun in Chapter 1. I will then be free to continue on to Chapter 3 to discuss in detail a case of handcoding in computational models of analogical cognition.

Making the handcoding critique useful for cognitive science requires defining *computational cognitive modeling*. As the name itself implies, there are three subjects to be addressed. I will first discuss what a *model* is and what explanatory role it plays in a scientific investigation (Section 2). Recognizing the role computational cognitive models play *as scientific models* is crucial: although models may always have some unavoidable aspect of handcoding in them, understanding the role of a model in explanation will show in what sense something has been handcoded appropriately or inappropriately. I will then give an elaborated definition of *computationalism*, the research methodology which employs computational models as devices to aid in explanation (Section 3). And finally, I will discuss the basic minimal set of criteria for the ontological status of *cognition* according to computationalism (Section 4). This discussion will establish the foundations for what computational cognitive modeling is so that framing the handcoding critique in terms of computational cognitive modeling will show exactly what kind of work the handcoding critique can accomplish as a methodological tool for theory building and analysis.

2 - Scientific Models

The role scientific models play in empirical enquiry is indispensable. This has been particularly shown in the history of science, as we can now look back and see the development, and at times, the replacement of fundamental views of how it is believed that nature works (Kuhn, 1970). One might ask how such replacement is possible given that fundamental views of nature are not directly comparable solely in terms of plain observation. While Kuhn's answer suggests that it is ultimately a matter of one discursive community outliving the other, "older" view, we can also look to the logic of the scientific use of models to see why the eventually dominant view, in retrospect, can be seen as more powerful and more satisfying.

Models give us the possibility of empirical access to mechanisms which operate "beyond the veil of perception," providing the forum for which we can consider how the world operates even though we cannot directly observe the underlying mechanisms of such operations. We could never come to an understanding of such underlying mechanisms based solely on observation. Through the use of models, we build pictures of how the behavior of all the kinds of independent phenomena we observe might be related to one another by underlying law-like mechanisms. These models may then be compared to one another and to our masses of collected observations of the world to see which gives us the "best" picture of how these independent phenomena could be related.

(I will not go into the details of the debate over whether or how such theories get closer to "how nature really is."²⁰ However, an understanding of what a model is will be crucial for filling out the idea of how the handcoding critique can be used in empirical enquiry in computational cognitive science as part of the machinery for developing better models, and as a consequence better explanations.)

Models also play a crucial role in computational cognitive science. With the rejection of behaviorist explanations, cognitive science allowed for legitimate empirical enquiry into the mechanisms which underlie the observed behavior manifested by intelligent agents: the "black box" of the mind/brain is now no longer considered impenetrable. Several developments allowed for the possibility of empirical access to these "hidden" mechanisms — two in particular. First, there was the theoretical work in mathematics that culminated in Turing's formalization of the notion of computation and the subsequent developments in technology which led to the invention of the programmable computer. These discoveries were very important for cognitive psychology and cognitive science in general because they established a feasible medium by which proposed notions of how the mind/brain works could be actualized in working models, opening these theories up to more rigorous empirical enquiry. What these discoveries amount to will be the subject of Section 2.2.

²⁰ In my discussion I am clearly keeping the faith that some such story does exist and the justification is sound. The current popular argument is that when comparing rival hypotheses (proposed rival models), the one that is chosen is the one that *best* explains the phenomena. 'Best' is in turn cashed out as that model which reduces (or shows an underlying law-like relation between) the greatest number of independent phenomena (and such reduction could include fitting in with other existing models which are also highly confirmed). This notion is central for the position of *convergent realism* (as well as some other varieties of realism) in the philosophy of science (Aronson, Harré & Way, 1995).

While these discoveries were being made there was also the development of key concepts in the philosophy of science which served to illuminate the logic for the proper use of scientific models in the investigation of phenomena which lie beyond direct observation — the second important development. The following discussion of models and their role in science is a “skimming off the top” of the large body of literature in philosophy of science and related disciplines which worked to develop the scientific use of models. I will not be adding much in the way of new insights here. However, doing this summary is nonetheless very important: I want to make these developed notions of a model for science in general to be useful for cognitive science in particular. The goal of this section (2.1) is to build this analysis of scientific models as a tool to be used in the evaluation of computational cognitive models under the handcoding critique.

2.1 - Some preliminary distinctions

Before going into more detail concerning what models are, I should first make some preliminary distinctions to clarify terms. The first is between two senses of “scientific model”: theoretical models and model exemplars. Overall, science could be said to be in the business of developing *theoretical models*. As Giere (1997, p.24) puts it, “a theoretical model is part of an imagined world. It does not exist anywhere except in scientists’ minds or as the abstract subject of verbal descriptions that scientists may write down.” A theoretical model is intended to be a description of how the world works. Giere’s description of a model as part of an imagined world is thus an important insight: models propose possible ways that the world might be. The question then is whether the model pictures a possible world that is similar to (or the same as) aspects of the actual world. Giere is also correct in highlighting that theoretical models by and large exist as a collection of things, including the scientist’s imagination and the discourse in which she presents her ideas.

This discourse, however, includes more than verbal description; it also includes *model exemplars*. Model exemplars include things such as scaled-down airplanes (scale models), analogies (analog models; e.g., the DNA molecule is shaped like a spiral staircase), or computer programs (a certain kind of scale model in which the term “scale” refers to kinds of computational states and defined functions instantiated in different mediums). These are all actual physical things. The discourse presenting a theoretical model may include these physical models, which are used as exemplars to help in communicating the researcher’s theoretical model. These exemplars help the researcher in the task of making reference to aspects of the phenomena in question, and to demonstrate how underlying mechanisms function — all of this is part of the task of making scientific discourse public so that communication can take place.

Thus, when we talk about the scientific theory that a researcher is proposing, we are ultimately referring to the theoretical model (or models), even though such discussion will often surround one of these physical model exemplars and how it is to be interpreted as representing nature. Why is this last point important? Because when we go to consider a computational model (in fact, any model), we have to understand that it is proposed to rest on a background of theoretical assumptions and assertions, some explicit and some implicit (and perhaps it is part of a whole family of models and corresponding theories; consider, for example, the variety of models and theories aimed at different

levels of phenomena and kinds of explanation, yet all related as part of the general theory of Darwinian evolution). This again brings up the point made earlier (in Section 3.1.2 of Chapter 1) that a running program is not itself an “existence proof” in the sense of providing an explanation simply because the program is able to accomplish some task. An explanation is *of* some phenomena (which for computational cognitive modeling is cognition), and just because some computation in certain circumstances can produce some desired behavior does not entail that “cognition in the wild” works that way — a working program may only constitute *one* of a number of solutions to a problem, and it may in fact be not at all similar to how naturally occurring organisms have to solve the same problem. Computational cognitive models are parts of theoretical discourse aimed at explanation, and so do not have some simple, purely objective existence. And more importantly, models do not themselves automatically provide interpretations as to their structures and what in the world they are proposed to correspond to. This will be precisely the issue in Chapters 3 and 4, when it comes to considering how “representations” in a computational model are to be interpreted.

In this next section I will be talking about models in general (including theoretical models and various kinds of model exemplars) in order to lay out the logic of the use of models in explanation; that is, how it is that models can be *about* the world.

2.2 - Models

Realist approaches to the philosophy of science, in reaction to logical positivist views of explanation, have in the past half century worked to introduce and develop a “proper treatment” of the role of scientific models as being a central figure in explanation (notably: Aronson, 1984; Aronson, Harré & Way, 1995; Black, 1962; Campbell, 1957; Giere, 1988, 1997; Harré, 1970, 1983, 1986; and Hesse, 1966, 1974). As a result, much is now understood about the roles models play in explanation, and this in turn has been supported by many now well-established cases in the history of science of the use of models in providing explanations of particular natural phenomena (e.g., analyzing the analogical model of “Niels Bohr’s atom as like a miniature solar system”; or the explanation of the relationship between the Boyle-Charles law of gases and the Newtonian mechanical model of nature as comprised of moving atoms to produce a view of gasses as collections of moving atoms, which in turn explains the relationship between temperature and mean kinetic energy). The purpose of this section is to introduce a brief summary of these developments in the various roles scientific models play in the logic of explanation.

To start with, it is generally agreed that to *explain* something is to somehow make it so that it can be understood. This is certainly not a technical definition, but we can at least agree that when we have an explanation we have: (1) some thing (a model) by which we can in some way observe (or imagine) the internal workings of something we want to understand; (2) we have developed a vocabulary to describe such workings; and (3) we have observed (or imagined) our model to behave consistent with the way we think it should. All three of these things lead us to a greater understanding than situations in which we lack these things. As Harré (1983, p.69) puts it, we can come to understand something, “either by finding an illuminating analogy to the phenomena whose character we do not understand, or by our ‘exposing a hidden mechanism’ the workings of which

inevitably result in the phenomena that required explanation. An explanatory theory may depend for its acceptance on the success of its analogies, or on the plausibility of the mechanism it postulates, or in many cases on both.” This idea is the start of the relationship between models and explanation.²¹

In explaining what a model is, Aronson (1984, p.67) notes that it is important to first highlight the similarities as well as the differences between the use of a ‘model’ in science as an explanatory tool and the technical use of the notion of a ‘model’ which occurs in symbolic and mathematical logic (a point sometimes confused in artificial intelligence “models”). First, the differences: In symbolic and mathematical logic, a model is a set of objects that satisfy a set of axioms of a formal system — here the model provides an ‘interpretation’ of the system. So, for example, if the mathematical system of geometry is strictly axiomatized, it would be determined that a variety of mathematical objects, such as a “point” or a “straight line,” satisfy those axioms. Models in science, however, provide more than just an interpretation of “what” is in a system; a scientific model allows us to transfer properties from the model to the system modelled. Scientific models are a way of picturing aspects of nature, such that the behavior of nature is not just predicted but we can see *how it works*; and we can see this in the model even if those parts of nature can not possibly be actually observed.

On the other hand, the logical as well as the scientific uses of models both start from the same basic idea: that the relationship between the model and the system modelled is that of *similarity*. The two systems (the model itself and the system modelled) have a common structure, which is technically referred to as a *structural isomorphism* between the two systems (Aronson, 1984, p.67). The fact that the two systems have a structural isomorphism entails that even though the two systems are distinctly different, in some aspect (or aspects) they share in common the same laws of behavior or “qualitative structure.”

In this way, a toy airplane may serve as a model of an actual airplane because there may exist a structural isomorphism between the toy and an actual jet airplane with respect to overall geometrical and aerodynamic features; or the toy airplane might have a further structural isomorphism with a jet by being made of a material that gives it approximately the same relative structural strength as the aluminum alloy found in the real jet. These particular structural isomorphisms may prove useful in, for example, testing the aerodynamics of the plane architecture, or its structural integrity, without having to actually put a real jet into extreme conditions. This is possible because certain aspects of the model will be (under the proper contextual conditions) the same as those of the system modelled. Computers can also be used to model or simulate the behavior of various physical systems because the instantiated “laws” governing behavior in the computer model are isomorphic to the laws of the system being modelled.

Also important to note here is that I use the term “laws” generically to refer to any kind of forces, constraints or boundary conditions that govern the behavior of nature

²¹ Note: a model is not the same thing as an explanation. To explain something is to somehow show what it is or how it works. A model (namely, a theoretical model, a part of the communication of which might be a model exemplar) may be employed in an explanation as a description of what something is or how it works. But the model by itself does not constitute an explanation; again, models do not provide their own interpretation — the interpretation is an additional aspect, as I will discuss below.

consistently and regularly within a certain context²², not just the kind of laws we might be inclined to think of in terms of classical physics. These kinds of laws of behavior include the organizational mechanisms involved in establishing biological structures, such as single cell or whole organism metabolisms; or the variety of organizational principles which govern evolutionary selection processes; or the boundary conditions for the formation of clouds.

Isomorphism is just a start to uncovering the role that models play in scientific explanation. The key power of models as tools in explanation is that an analogy may be made between the model and phenomena to be explained. The model is said to be something we understand in the sense that we are familiar with its internal workings. Thus, "...by means of [the] analogy, the language and properties of the model are transferred to the system modelled, enabling us to think of the unfamiliar in terms of the familiar. More important, *new properties* are ascribed to the unfamiliar phenomenon by means of its analogy to the familiar system" (Aronson, 1984, p.68).

Of course, not all analogies will be useful in explanation. For this reason, a distinction has been made between "formal" and "material" analogies (Aronson, 1984, pp.71-72). Formal analogies are those in which the two systems resemble one another in their behavior, but we could not use one to explain the behavior of the other — somehow the internal workings of one do not match the internal workings of the other. Aronson (1984, p.72) offers as an example the surface similarity of the equation of motion that describes a swinging pendulum and the equation of an oscillating electrical circuit; both equations have the same form, so the behavior of one could be used to mimic the behavior of the other, but it has yet to be found how pendulum behavior can explain electrical circuits, or *vice-versa*. These are not the kinds of analogies intended for explicating the scientific notion of a model. Material analogies, on the other hand, not only hold an isomorphism of behavior, but the key properties of the model are *replicated* in the modelled system (Aronson, 1984, p.73). Scientific models expand our theoretical vocabulary for the phenomena we wish to understand. It is this kind of ascription of a new and "richer" set of properties from the model to the phenomena that funds an explanation.

Finally, it is important to note a key feature of analogies themselves: an analogy made between two things entails that there are ways in which the two things are similar (alike) *and* dissimilar (not alike) one another. So, in explicating a model's relationship to a phenomenon, it is important to make clear how the model is positively analogous to (i.e., 'like') the world and negatively analogous to (i.e., 'not like') the world. This helps pick out exactly what it is that we see in the model that is also the case in the world. In this way, the positive and negative analogies together help to pick out what "slice of nature" the model represents. It is the task of the theorist using the model to supply the criteria for filtering positive from negative aspects of the analogy (Aronson, 1984, p.71; Hesse, 1966, pp.57-129).

²² That is, given a context of certain conditions, those forces or constraints that govern behavior will be the same. This assumes a theory of contextual identity (see Aronson, 1984, Aronson, Harré & Way, 1994, and McClamrock, 1995, for further justification and development of this idea and related issues).

2.3 - Measuring Theories

So far in my discussion of models I have been talking about *kinds* in nature and models as if their definition and identification were simply given. In fact, this is not the case — such claims are additional theoretical assertions, as introduced in Section 3.1.2 of Chapter 1. The positing of their existence and how to identify them constitutes two additional (although intimately related) theoretical assertions.²³ First, there is the theoretical assertion that certain kinds exist: the positing of some ontological framework which stipulates what kinds exist, what their nature is, and how they are related. For purposes here, I will not go into detail about the philosophy of kinds (I refer the reader to Harré, 1986, Part II, pp.97-144 for a nice discussion and development of the metaphysics of natural kinds). But with the ontological status of *kinds* given or asserted, a general set of criteria exist for their identification and quantification.²⁴ This is the second assertion: a *measuring theory*.

In what sense do measuring theories exist? Consider three cases. First, we take a thermometer and look at its current state. The thermometer's mercury fills the small glass tube up to a mark of 20° Celsius. What does this show us? Assuming the thermometer isn't somehow malfunctioning (e.g., it isn't cracked), and it's being used in the proper conditions (it has been in the room long enough for the thermodynamic state of the mercury to have reached equilibrium with respect to outside environment), and it is indeed a proper thermometer (it hasn't been "miscalibrated"), we can claim that the room is 20° Celsius. But as can be seen, there are a number of assumptions which accompany this claim, and all of these in turn rest on some background theories. Most important of which is that mercury behaves a certain way (expanding in volume in warm temperature and shrinking in cool temperature). There's also a straight-forward convention, that "20° Celsius" marked on the present position of this thermometer will represent a state of the room (after sufficient time) corresponding to a certain local amount of mean kinetic energy (temperature), and will do so in similar conditions in other places. In this case, the place of the measuring theory (I'm referring to all the parts involved in the assumptions of how to "read" a thermometer and what that "reading" means as parts of the measuring theory) is fairly straightforward. And even though differences in what the thermometer may represent can arise, the behavior of the

²³ These are referred to as *auxiliary hypotheses* because they are hypotheses that accompany the basic *theoretical hypothesis*, which asserts that the proposed model does "fit" the world. Again, this drives home the point that a theoretical assertion as a whole is comprised of many independently distinguishable claims and assertions. The purpose of being so methodical in picking out these aspects of theoretical claims is that we need these distinctions in determining where and how handcoding occurs, and what impact it has on a theory (i.e., which claims fall because of handcoding, and which remain intact).

²⁴ Note that not all kinds need to be necessarily detectable or measurable; many exist as theoretical entities and some are in-principle not possibly measurable. Nonetheless, we may still have good reason to posit their existence, *and* they do still hold certain criteria for how they are supposed to behave and affect other kinds, depending on their ontological status. Here I will only talk about those that are measurable, as the characteristics of strictly theoretical entities are subsumed in the description of measurable kinds, and this is what is important for the coming discussion of interpreting measurable physical states as being mappable to states and functionally prescribed state changes in a computational description.

thermometer, and how to read its state, is more or less a fact common to all observers (who know the convention).

A second case, which is more revealing of the role of a measuring theory, can be found in the history of the discovery of sub-atomic particles (Harré, 1983, pp.153-154; Hanson, 1963, gives several examples). What is revealing about this case is that even the possibility of a common observable fact can be brought into question. According to modern physics, electrons (negatively charged particles) follow a characteristic path in a magnetic field. In the early half of this century, the question arose as to whether there were any particles of the same mass but positively charged. Anderson reported finding a track of a particle, the length of which strongly suggested that it was of electronic dimensions. The question was then put forward: was the picture of a negative electron that picked up energy in the magnetic field (having a track of lesser curvature at the end of its flight) or was it a positive electron which lost energy and so curved more sharply under the influence of the field? When considered at the level of physics (namely, in which this observation might be considered as evidence for or against Dirac's theoretical analysis), there were clearly *two* facts possibly represented by the picture, not one: "if the track was made by an electron of negative charge the photograph represented one fact; if it had been made by a positron the photograph represented quite another" (Harré, 1983, p.154).

Any observation is likewise taken as being a particular kind depending on the background conceptual framework:

“ ... to accept that the photograph is indeed a *picture* of the track of an ionizing particle a very great deal of theory must be presupposed, and not only physical but chemical theory too. Only relative to a whole cluster of theories which everyone involved shared does the photograph exist as a collection of data at all.” (Harré, 1983, p.154)

As Harré goes on to mention, even taking the picture as being *of* anything depends on being embedded in the conceptual framework of sub-atomic physics; to the lay-person of the 1920's and 30's, the picture would probably have no meaning at all. The example of theoretical sub-atomic physics brings out this dependence on background theory and conceptual framework since how to take an observation as being of some kind is not as well established (not as well-confirmed) as, say, watching the sun rise each morning and set in the evening and taking it as a phenomena produced by the earth rotating on its axis; nonetheless, the same kind of dependence is there in both cases (of course, in one case,

we have much more confirmation, and therefore much more certainty in our assertion).²⁵

A third and final example sheds light on yet another facet of how measurement theories exist: the way a measuring theory carves up nature may pick out particular information while ignoring others — and they do so based on a particular framework for representing scientific knowledge. The curious case of Mendel’s “fraud” makes this point (Harré, 1986, pp.170-171). The Austrian botanist, Gregor Mendel, performed a series of experiments on the breeding of peas to determine how genetic inheritance worked. It is unclear exactly what methods he used, as his notes were destroyed shortly after his death. What makes his achievement questionable and curious is that he somehow managed to roughly predict a general principle of genetic inheritance, while it has since been proven that based on the supposed method he used, his chances of obtaining actual data which would support his conclusion are about 1:30,000. How did he do it, and why might he have drawn false conclusions?

Root-Bernstein (1983), has proposed a possible solution to the mystery which makes it not so much a “fraud.” He argues that in Mendel’s time there were two fundamental ways of viewing nature: (1) the *biological view*, which saw individuals as varying continuously in all their common characteristics, and (2) the *statistical view*, which was based on counting the frequency of occurrence of attributes within a population. In the statistical view, to be “countable” required being discrete, not continuous. Thus, an individual must be either green or yellow if a head count of green peas and of yellow peas is to be possible. As Harré (1986, p.170) puts it, “greenish-yellow things are an embarrassment. In order to apply statistical methods to biology it was necessary to solve ‘the problem of assigning continuously variable characteristics to discrete categories’ [Root-Bernstein, 1983, p.279].”

Root-Bernstein puts the point succinctly: “the ‘reality’ of nature confounds the ‘ideality’ of classification” (p.280) — at least a classification system that yields statistically treatable measures. So what Mendel must have done (if this story is true), is *assign* the 7% of peas which did not unambiguously fit his predetermined discrete categories to discrete categories which were already well filled with specimens. According to Root-Bernstein, “what in fact Mendel published was not a ‘real’ description of his peas, but his perception of how those peas could be categorized into ‘ideal’ discrete groups” (p.282). “[Mendel’s categories were]... theoretical constructions *by the help of which* nature was segregated... Mendel’s analysis depended on the use of relatively a priori categories which emerged not from his experience of nature, but from the

²⁵ Just to be clear, none of this is to suggest some form of strong relativism. While scientific facts are constructed, they are done so in the practice of comparisons between rival hypotheses and models, in an empirical framework. Their confirmation is a part of a whole emerging conceptual framework. So confirmation may be relative such that all hypotheses may be subject to revision, but eventually so much background conceptual framework will be at stake that revision comes at a great price. For example, challenging that our observation of the sun rising and setting is not about the earth turning on its axis would likewise challenge our whole system of belief about our solar system and most of astronomy, something we are not going to want to do unless presented with a greater and better confirmed framework — something which doesn’t seem likely to happen. Aronson, 1984, p.150, notes this point in his discussion of the possibility of crucial experiments, and points out how Kuhn overlooked this in his analysis of change in science as merely discursive communities outliving one another.

exigencies of the method he had chosen for studying the phenomena” (Harré, 1986, p.171). In short, Mendel employed a measuring theory which was derived from an entire theoretical framework for a kind of scientific study of populations.

Analogous cases certainly exist today (e.g., the psychometrics framework used in the study of categorical perception; Harnad, 1987). Such frameworks have had successes and failures, and much has been learned about when it is and is not appropriate to make use of such frameworks; much of this hinges on the nature of the phenomena in question. An important point to understand here is that in some cases, it is not the particular study itself which needs to be brought into question, but rather, the whole theoretical framework from which a particular measuring theory used was derived. (An analogous situation exists in considering the nature of representation, as I will discuss in Chapter 4.)

A measuring theory is therefore the set of assertions which accompany a general proposed hypothesis of how it is believed the world works. The measuring theory regards how some phenomena in the world is to be taken as being an instance of some kind. With respect to a model, a measuring theory thus has two parts. The first is what gives a model its *semantics*: assigning what ontological kinds correspond to what aspects of the model. The second part of the measuring theory, which is concerned with the model (which has been given a semantics) and the model’s relation with the world, plays the important role of generating *predictions* from a model. Predictions include more than just “what will happen in the future” — they should be taken more generally as what the model’s behavior and entities are to be interpreted as being instances of; i.e., what *kinds* of behavior and entities they are. The measuring theory role in terms of the world, however, is not easily separated into two roles, but instead takes the observations made in the world and interprets them as being certain kinds — such judgements produce *data*. Data, the complement to predictions, are then the collection of judgements made as to what kinds certain observations are taken to be.

With this notion of a measuring theory, along with the above general discussion of models, I can now draw a Figure (2.1) which depicts the relationship between models and the world, and the role that measuring theories play in those relationships. (This picture is adapted from Giere’s analysis of models; Giere, 1997, p.36; note: I have reversed the order from Giere’s original “Model-on-the-right” scheme):

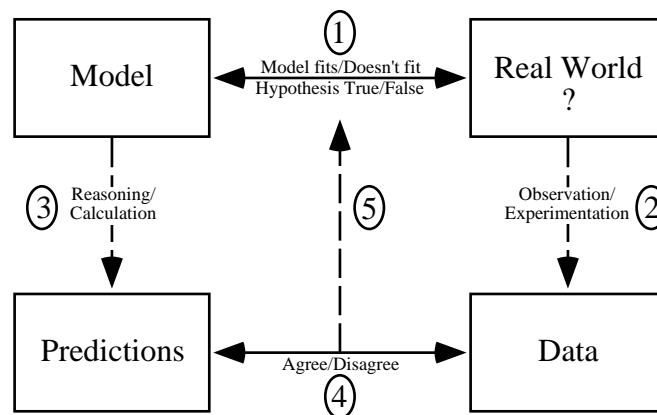


Figure 2.1 - Giere’s Analysis of Models

Link (1) depicts the focus of model's comparison with the world. This link represents the *theoretical hypothesis*, which asserts that the model does provide an accurate picture of the world. The goal of an empirical investigation is then to test this hypothesis. As the discussion of measuring theories made clear, models themselves cannot be directly compared with the world; the world does not provide its own interpretation, and neither do models in themselves — a measuring theory turns uninterpreted phenomena into meaningful data.

Links (2) and (3) represent the application of a measuring theory to provide an interpretation for particular observations made in the world and structures and processes found in a model (link (3), however, is really only the second aspect of a measuring theory with respect to model semantics — this figure here assumes the model's semantics have already been determined by the first part of the measuring theory). Links (2) and (3) differ in that link (2) is often a matter of actually physically measuring the world (usually done through some sort of experimental apparatus), whereas link (3) might be either reasoning what would occur based on how the model is intended to work, or by actually “running” the model in a “simulation.” Despite this distinction between links (2) and (3), it is important to stress that the same measuring theory is employed in both cases; i.e., the same kinds of interpretations will be made of the subject matter.

Application of the measuring theory on the particular procedures appropriate for garnering an interpretation from the real world or a model yields data and predictions. It is then the data and predictions that can be compared²⁶ to get an idea of how well the model (based on the background of a particular measuring theory and subsequent interpretation) fits the world (similarly interpreted). This comparison is link (4). Once this comparison is made, we now have information regarding the status of the theoretical hypothesis (link 1) — this information is represented in link (5).

As cognitive scientists evaluating some proposed computational model, it is important for us to understand what theoretical framework that model fits into, including its measuring theories. These are what pick out what *kinds* of entities and processes in the model correspond to what are proposed to be the *kinds* in the natural cognitive systems that we wish to explain, and the criteria for how these kinds are identified. Only once this has been done can the model be properly evaluated. And, as would be expected in any young science, many of the disputes in cognitive science are over whether particular measuring theories are valid and properly applied; many of these issues will be fought at the conceptual level, rather than entirely empirical.

²⁶ The logic of the possibility of this comparison follows from the fact that, given that the same measuring theory has been employed in the production of data and predictions, they are both on the same “level” — i.e., the same basic method of interpretation has been applied in both, even though the particulars of the procedure of how to collect data and the particulars of aspects of the running of the model may be different. What's appropriately different depends on the explanatory task (discussed in the next section) to which the model is being put. In the upcoming section discussing supervening machines (Section 3.5), I will spend some time making more clear how levels of physical operation which instantiate a supervening machine might not play an important explanatory role in the semantics of the model that the supervening machine represents.

Just as with models in general, the role of the measuring theory in computational models also has two parts. The first indicates what kinds of computational structures are instances of what kind of behavior, entity, or process in the real world. This part will be referred to as the *theory of interpretation*, which I will describe in more detail below. The theory of interpretation is what stipulates the computational model's semantics (Figure 2.2 depicts this relationship):

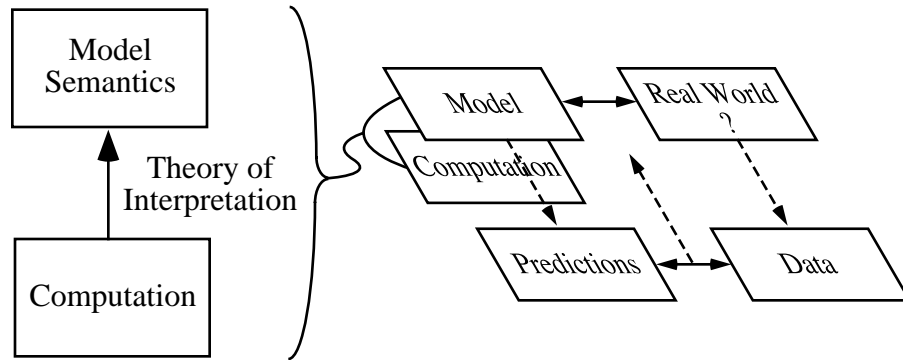


Figure 2.2 - The Role of the Theory of Interpretation

The second part of the measuring theory, once the model's semantics have been determined, plays the same role of then garnering predictions based on the model, as discussed above; in computational models this usually the observations based on the behavior of the running program.

2.4 - The relation between models, theories and the world

In order to finish this discussion of what models are, I will characterize the variety of features of a model which determine the relationships a model has with the world in the context of a theory proposing an explanation. These relationships depend on the background theory of which the model is a part. Most of these relationships have already been introduced, but here I will use an example of a map of the layout of a city to make them more concrete. A map is a good example because it is something we are all familiar with, and it bears the same kinds of relationships that any other model might have — a map is, in fact, a kind of scale model. In parallel, I will discuss the particular manifestations these features of models in general make in computational cognitive models in particular. Although I have not specified the other two aspects of computational cognitive modeling (computation and what computationalism takes cognition to be), this discussion will set the stage for the eventual completion of these definitions.

Summary of the features:	General Question	Issues
(1) The Model's Intended Use	What is the use of the model in the explanation?	Explanatory Task Explanatory Goal
(2) Relation between model and the world	How does the model relate to the world?	Structural Isomorphism: Differences & Similarities
(3) Semantics of the model	How is the model to be read / interpreted?	The background theory interprets the model; Identity
(4) Accuracy	How accurate does the model need to be?	Efficiency vs. Explanatory Depth

(1) The model's intended use: explanatory task and goal

The first thing to note is that the map is intended to be *used* a certain way. This means that the model has a particular *explanatory task*, which includes what task the explanation serves (i.e., what are we using the model for?), and thus what the *explanatory goal* is (sets the criteria for what it means to have been successful at the task of explanation). Task and goal are clearly closely related and often treated the same, but I am dividing the task (what is to be accomplished) from the goal (what counts as the task being accomplished) to make the point that many models may have clearly defined explanatory tasks, yet fail to reach their goals.

In the example, the map is to show how to navigate from one building to another via roads (the explanatory task of the map), and is thus considered to be a model of the layout of a city — a map has satisfied this task (reached its explanatory goal) if it can be appropriately used to navigate. Thus, the map is intended to be similar to the actual layout of the city (i.e., to *explain* the layout of the city).

Analogously, computational cognitive models are also intended to serve particular explanatory tasks. A variety of explanatory tasks are possible: e.g., to demonstrate necessary conditions required to instantiate some intelligent function, to show us how a mechanism functions in a variety of situations, to show us that some function is implausible as an explanation of a cognitive phenomena, to serve as a high-level sketch of how it is believed a phenomena behaves, etc. These in turn designate particular kinds of explanatory goals — in the cases just mentioned, they are all generally intended to be some working model which can somehow demonstrate some behavioral capacity (or not) — but the criteria they demand are not the same: a “sketch” of how a phenomena should behave, e.g., should not be judged solely based on precision; nonetheless, it can still be a valuable tool as an intuition pump, and to that end be successful — it therefore shouldn't lose its “model-hood” just because it fails at precision. Understanding the explanatory task (and subsequent goal) to which a model is put is crucial for accurate analysis of the model.

(2) The relationship of the model to the world: similarities and differences

The next aspect of the map concerns in what sense the map is similar or dissimilar to the actual city. This invokes the idea introduced above of an isomorphism relationship between the model and the system modeled. What are taken to be the “appropriate” *similarities* and *differences* depends on explanatory goals as well as particular commitments made to the *identity* (ontological status) of a particular phenomena to be

modeled. Thus, the map is intended to somehow have or show a layout similar to that of the actual city. It is here that there are also the beginnings of commitments to particular measuring theories.

The map is *similar* to the actual layout of the city in that the relations between the marks on the map for buildings, roads and rivers correspond roughly to the actual relations between the real buildings, roads and rivers of the city. This highlights that the explanatory goal is to capture the same relations, such that the relations between the marks on the map, which correspond to buildings, roads and rivers, are the same as (relatively: have the same relative proportions as; i.e., *isomorphic with*), the relations between the buildings, roads and rivers of the city. In this sense, these relations are taken to be the identity of the particular phenomena that the map is said to model. Since computational cognitive models generally have the explanatory goal of specifying or explaining the function or behavioral capacity proposed to underlie some cognitive phenomena, they should somehow embody that function or have the capacity to produce some behavior (or internal processing, etc.) similar to that of the actual phenomena.

The map is also crucially *dissimilar* in that the actual city consists of buildings and houses made out of concrete, steel, brick, wood, etc., the roads are comprised of pavement and painted traffic lines, and the natural landmarks of the city are comprised of geological and biological formations (trees, changes in elevation, rivers of flowing water, etc.). The map, on the other hand, is simply paper with ink marks on it. The map and the city are made up of different things. It is assumed here that for a proper map to achieve its explanatory goals, these details are not important; or better, which details are to be included is a function of the use of the model.²⁷

What position a cognitive scientist takes on the issue of the nature of cognitive phenomena will likewise determine what is allowed to be dissimilar. For example, the functionalist (e.g., Armstrong, 1981; Putnam, 1967) argues that the essence of cognitive phenomena is not determined by the physical materials involved in the makeup of the system that instantiates the function. Instead, the determinates of the identity of cognitive phenomena are a matter of the functional relationships within the system itself and between the system and the environment it is in. Thus, it doesn't matter if the function is instantiated in a biological-based architecture, a silicon-based architecture, or in Tinkertoys™, as long as it can maintain the defining functional relationships. Searle (1992), on the other hand, would argue that the model has to be biological neural-tissue in order to *really* be cognitive.

²⁷ Also important to note is that there is a third kind of analogical relationship in modeling analogies: a *neutral* analogy. The neutral aspect of a model's analogical relationship to the world involves the properties of the model and system to be modelled that are not yet examined — these are the aspects of the model and the world for which it has not yet been determined what the model and world's relationship is. Neutral analogies play an important role in furthering ongoing scientific investigations. And just as with positive and negative analogies (ways the model is similar and dissimilar, respectively), neutral analogies must be discovered (or rather, uncovered).

(3) The semantics of the model

Of course, to read a map *as* a map of a layout of a city requires a set of social norms: rules or conventions for translating kinds of map-directions into kinds of directions for actual navigating in the world. (Included in these conventions are the particular measuring theories that are conventionally accepted: e.g., for picking out “buildings” [certain-sized structures made of certain kinds of material] as buildings, “streets” as streets, etc.) Likewise, there are sets of conventions for understanding models and what they are explaining. Understanding the appropriate similarities and differences, as just discussed, is a crucial part of understanding the model. In maps, these conventions are usually explicitly laid-out in a “legend.” The legend designates which aspects of the map are paired up with which aspects of the world, as well as the relative scale of map features to corresponding features of the world. The legend of a map, then, sets what the structures of the map and the world are that are dissimilar in kind, but correspondent (mapping *kinds* from maps to *kinds* in the world) — again, legends get their job done because there are a set of conventions which exist which set how the legend is to be interpreted.

In scientific models, the analog of a “legend” is provided by the measuring theory, and other background theoretical assumptions for which the model is being used as an exemplar. Parts of the background theory help specify which aspects of the model correspond to which aspects of the world, and the measuring theory helps in the task of actually picking the kinds out. In general, I will refer to the aspect of the background theory which provides the correspondence specifications between kinds in the world and kinds in the model as the *semantics of the scientific model*. (And for clarification, for both the map and the model, the aspects that should *not* be interpreted as being *differences* in kind are those that the model *is* intended to capture — i.e., those which are intended to be isomorphic.)

It is important to note that it is here that the issue of correct versus incorrect handcoding is partially decided. As alluded to above, there is a sense in which any model is going to involve handcoding — that’s what it is for something to have an analogous (similar, but not same) relationship to something. (For example, with respect to the map example, “handcoding” in a general sense might be the use of squares to represent buildings.) Depending on explanatory goals, certain kinds of handcoding is legitimate. For example, with the map it is OK to specify that a building will be represented as a rectangle because the map is intended to only capture the spatial relationships between objects in the world — in this sense, the particulars about the building drop out of consideration in the map; the only thing that is important is the rectangle’s spatial relationship with the other objects on the map relative to the relationships of objects in the world. Here, the only requirement is the convention that reliably picks out the proper correspondence relationship: i.e., that rectangles in the map are always taken to represent buildings in world.

Having such a convention for picking out correspondence, however, does not always ensure that handcoding is legitimate — particularly if it is the identity of phenomena to be modelled that is contentious. In the case of the map, it is hard to picture how this problem could arise. But suppose someone were to challenge that buildings really aren’t what the map makers thought they were; this could affect the legitimacy of the map as a

good model of the spacial relationships between objects in the world. It may turn out, for instance, that buildings are really just mirages. Thus, as far as getting around the city is concerned, it's no longer important, and maybe even detrimental, to represent buildings as "objects to be gotten around"; for these purposes, buildings aren't legitimate "objects" whose relations with other solid objects of the world need to be considered.

(4) Accuracy

A further issue concerns the *accuracy* required so as to consider the model an appropriate explanation. This, of course, is also relative to explanatory goals. The map, for its intended navigation purposes, doesn't have to be perfectly accurate to meet its goal of being a useful map of the layout of the city for getting from one building to another. The map doesn't have to include every tree, the location of every rock in a river bed, or even the exact locations of the foundations of buildings, or whether the buildings have windows, etc. In fact, the map may not even have to get the cartesian relations of the streets precisely correct, as long as it is still possible for the map to be used by a navigator to get around — to distinguish buildings from one another and which streets are which, and that these still have the same topological relationships. In fact, were there a map that was so complete in all its detail that it accurately described a vast number of properties of the mapped area that were irrelevant to navigation, its specificity would defeat its purpose: to generalize and abstract (Gleick, 1987, pp.278-279). Again, what of a map should be accurate is decided in part by the explanatory goals to which the map-as-model is being put.

The issue of accuracy in cognitive modeling, however, is much more contentious — and this is one point where the analogy between my map example and general scientific models breaks down. In general, models are compared on the basis of which provides the "best" or "more accurate" explanation. It is generally held that which model comes closer to replicating the desired phenomena is the better model. However, particularly in the case of cognitive models, there is much disagreement, not only on what's considered to be "closer to the actual phenomena," but even on what exactly the phenomena is. Cognitive science is generally at a stage in which just being able to replicate some *kind* of behavior is much more important than how precise the behavior is replicated to that kind — and it may be the case that with respect to intelligent behavior, kinds are all that matter.

The issue of accuracy brings up yet another important issue which bears mentioning here: *efficiency* versus *explanatory-depth*. Hofstadter raises this point early on in his book (1995, pp.52-53), in his discussion of the SeekWhence model: there can be a large gulf between different research projects which on the surface seem to be in the same field. The difference, as would be expected, again depends on explanatory goals. On the one hand, there are those who seek to present a model of a mechanism that can perform some task as efficiently as possible. These are often considered to be on the "engineering" end of the explanatory spectrum. The explanations sought in these models are not necessarily for human-cognitive reality, but more importantly, seek to get a task done in the fastest and most economic way possible.

On the other hand, there are research programmes which seek to best explain how cognition occurs naturally. This is usually taken to be the domain of cognitive science

and all of its subfields — and, as Dietrich (1995, p.126) notes, this makes this branch of AI and cognitive science a branch of the more general field of ethology. Computational cognitive models in this domain will often trade-off immediate efficiency for eventual *explanatory-depth*, in the sense that these models seek mechanisms that hopefully capture cognitive structures which are indicative of more robust information processing in more complex domains, or can produce phenomena that match the kind of behavior we expect from “higher-cognitive” organisms. And, as the term “explanatory-depth” suggests, these models are often proposed as pieces of a more general set of models for a broader theory of cognitive phenomena.

This “explanatory-depth” can often be observed in models in which the domain in which the mechanism operates is very simple relative to the complexity of the behavior of the proposed mechanism. It is generally the hope that such a mechanism can then be extended to the more robust domains that “higher”-intelligent agents live in, and keep relatively stable and “intelligent” performance with little change to the actual architecture of the proposed mechanism; or the mechanism is intended to be integrated into a more complicated and robust model which is proposed to account for more complex behavior.

Therefore, as Hofstadter (1995, p.53) points out, in terms of the goal of explaining human intelligence, the most efficient program is not always (in fact, often not) the *best* explanation. For example, Hofstadter’s SeekWhence program, which searches and extracts (“discovers”) possible algorithms that may have created a sequence of numbers, would be misunderstood if it were compared only to a simpler algorithm, such as a plain breadth-first search, which might “solve” some sequences faster. The SeekWhence program is intended to not only discover the sequence, but do it by employing “deeper” rule-constructing mechanisms as part of a general research programme aimed at explaining high-level perceptual mechanisms. Here, the engineering concerns of speed and efficiency are pushed to the background, and the *kind* of mechanism that might be like perceptual mechanisms come to the fore.

The field of AI (in general) thus has a large internal split concerning explanatory goals. Handcoding issues are therefore also different depending on which kind of programme goals you are talking about. For the “engineers,” bad handcoding comes in the form of modeling aspects that might lead to inefficiency (perhaps brittleness in domain adaptivity) — but it is only relative to efficiency; the goal is to make the most efficient mechanism. For cognitive science, on the other hand, it will, of course, be relative to the explanations of the mind/brain. Issues of brittleness and plasticity may enter into the argument — often in the form of debate over what is innate and what is developed — but these are always, at root, empirical questions (and even the computational modeling paradigm, which I will discuss next, may be replaced by a more accurate descriptive language). The goal is to model natural cognitive performance.

I should mention here that there *is* an argument regarding efficiency that is legitimately used *within* the domain seeking to explain natural occurrences of intelligence: the argument from evolution. This argument holds that it is most likely the case that through evolutionary selective pressures, the mechanisms selected for will be those that tend to be more efficient (or better, “efficient enough to get by”). This, then, does bring the issue of efficiency into the realm of explanations of natural phenomena. But it should be clear that this is still distinct from the “pure engineering” perspective, as this argument is then dependent on the context of the naturally occurring evolutionary

mechanisms that would have participated in the production of the natural phenomena in question; the explanation is still essentially concerned with naturally occurring phenomena, not just strict engineering-efficiency concerns.

Confusion has naturally crept into both sides of the “race-for-intelligence” as, naturally, both sides often will have technology as well as theoretical insight to offer one-another. And it is not always the case that each only makes discoveries in its *own* domain. However, the crucial point to keep in mind is to make explicit both in understanding and in discourse which criteria are being applied in judging success: efficiency alone, or best explanation of how intelligence naturally occurs. And whether the ultimate destinations of either projects eventually converges is an empirical question. The research programmes I am most interested in, and for which I will be specifically aiming my notion of the handcoding criticism at, are those directed at the explanation of intelligence as it is manifested in nature.

3 - Computationalism

Now that I have introduced what a model is, and described the explanatory role it plays in a scientific investigation, I turn to define what computationalism is. This definition is necessary for establishing particular aspects of the kind of model used by computationalism in an explanation. In the process, I will also discuss the minimal set of ontological commitments computationalism makes to the identity of cognitive phenomena.

There are currently many versions of computationalism, each making particular claims about the details of what kind of phenomena cognition is, and how it is to be explained. Each of these versions of computationalism, however, shares in common several key foundational assumptions about what kind of model is to be proposed, what kind of phenomena cognition is, and how those models are to explain that phenomena. The purpose of this section is to develop the *minimal* set of foundational assumptions made by the various kinds of computationalism. This definition of computationalism follows closely that proposed by Chalmers (1994) and Dietrich (1990, 1994). Chalmers (1994) dubbed the version of computationalism he has developed as *minimal computationalism*. This is a good name; however, I cannot adopt it as I do not agree with key details of the “theory of implementation” central to Chalmers’ position, for reasons which I will discuss in the coming sections. For this reason, I will refer to the following version of computationalism as *minimal empirical computationalism*. Defining any version of computationalism requires starting with computationalism’s central hypothesis, the *computational hypothesis*, to which I turn now.

3.1 - The computational hypothesis

The *computational hypothesis* asserts the following: (1) the Church-Turing thesis is true (that the kinds of computations described by Turing-computation captures all that is computable) and (2) all cognitive processes are the computations of some function (and, therefore, that anything that computes these functions is itself cognitive). Before

expanding on the particulars of these two claims, it is important to note that this *is* a hypothesis, meaning that it is falsifiable — and this is an *empirical* issue. The hypothesis will have been falsified if it can be demonstrated that either of the two assertions are false: i.e., if it turns out that defining aspects of cognition are *not* “captured” by computation, or if it turns out that the Church-Turing thesis is false (that Turing-computation does not capture all computable functions) *and* cognition *is* the kind of computation that is *not* Turing-computable. (Dietrich, 1994, p.15)²⁸

The computational hypothesis requires expansion to fully flesh out its details. This will include characterizing what is meant by “Turing-computation” for an understanding of the first premise. Clarification of the second premise requires some further work. Chalmers (1994) has correctly identified that the plausibility of the computational hypothesis rests on two additional theses: the thesis of *computational sufficiency*, which claims that the right kind of computational structure suffices for the possession of a mind, and the thesis of *computational explanation*, which claims that computation provides a general framework for the explanation of cognitive processes and behavior. Discussing these two further hypotheses will explain the second premise of the computational hypothesis.

In order to establish Chalmers’ two theses of computational sufficiency and explanation, I will need to accomplish two tasks. The first is to establish a theory of *interpretation*²⁹, which will show how aspects of the physical world might be consistently interpreted as instantiating computable functions. The second task is to then argue for the plausibility that cognitive processes can be captured in a strong sense by this interpretation scheme so as to be described by a computation. Having established this gives good grounds to accept the truth of computational sufficiency for capturing cognitive processes. Likewise, it will serve to establish computationalism as a viable

²⁸ There is another possible set of affairs which could occur and would be a cause for concern for the computationalist (Dietrich, 1990, 1994): it might be shown that the human brain routinely solves problems which we know are NP complete (i.e., could solve certain kinds of problems which computationally require non-deterministic polynomial time to compute — these are problems that do have effective procedures [i.e., total solutions], but if the situation is even a little complex, it would take more time to compute than the time span since the Big Bang). If this were the case, then whatever our brains are doing is very different from computation as currently conceived. In fact, there is some evidence that certain biological structures in the retina of the eye do such problem solving in deciding which direction the eye should saccade. The key question to be asked here, then, is whether the computation that the retina is following constitutes an actual effective procedure which somehow solves an NP complete problem (i.e., solves the NP complete problem with *no* hint of what the correct outcome should be), or whether it is instead a product of evolutionary engineering that instead has imbued the eye with a set of “heuristics”— “rules of thumb” that do not guarantee absolute success in all cases — that are antecedently informed about kinds of solutions are likely to succeed (because of “evolutionary experience” with what heuristics have been successful in the past), in which case the eye cannot really be said to be *solving* NP complete problems. Even if it turns out the eye does somehow have a complete, non-informed “solution” to the NP complete problem it faces, this does not necessarily falsify *computationalism*; but this could then constitute evidence that our current theory of computation does not capture a kind of computation — one we have yet to understand.

²⁹ This theory is a combination of the basic spirit in Dietrich (1990), with special attention to several themes discussed so far in the development of models and measurement theories, and the general notion of the theory of implementation in Chalmers (1994).

methodology for the scientific investigation of cognitive phenomena (i.e., the empirical investigation of the computational *hypothesis*). Along the way, I will show how the computational hypothesis fits in with the camps in traditional philosophy of mind, which will serve to show the ontological commitments minimal empirical computationalism makes to the identity of cognitive phenomena. In the following discussion, I will refer to the general position of the “computational hypothesis” in minimal empirical computationalism as just *computationalism*.

3.2 - Turing-computation and the Church-Turing Thesis

The notion of a computation, and particularly its formalization, is one of the most significant accomplishments of mathematics. What is compelling about its formalization is that it seems to capture what many mathematicians have and do feel is the intuitive notion of a computation. I will start with an intuitive characterization of computation, building up to a formalized notion equivalent to that which was introduced by Alan Turing (1936).

The term *computation* denotes a step-by-step, mechanical process with a definite beginning and a definite end where each step (called a state) and each transition from one state to the next is finitely and unambiguously describable and identifiable” (Dietrich, 1994, p.7). To get a handle on how to characterize these states and transitions, I must invoke the notion of an *effective procedure*; this is an, “unambiguous, precise description of a set of operations applied mechanically and systematically to a set of tokens or objects” (Dietrich, 1994, pp.6-7). It is these operations that correspond to the transitions described in the computation. Thus, transitions will change the configurations of the tokens or objects from one state to the next. An additional aspect of effective procedures is that they must be mechanically and finitely capable of completion.³⁰ And finally, for every effective procedure, the initial state of tokens is defined as the “input,” and the final state is defined as the “output.” It is important to stress that the description of each state, transition, and total effective procedure is *finite* (Dietrich, 1994, pp.7-8) — this is important for keeping in line with Turing’s original description of computation (allowing for any of these to possibly be infinite would allow for a system stronger than Turing-computation).³¹ Given this, I can now define a computation as: the execution of an effective procedure.

A computation may also be described as computing some *function*. A function can be defined in two ways: extensionally, or intensionally. An *extensional* definition of a function is simply the set of all the pairs of inputs and corresponding outputs that would result if we were to take each given input state and perform the computation on them to

³⁰ Note here that I have not defined what “mechanical” or “mechanically” means; in the next section, this will be cashed-out in terms of corresponding to physical-causal relationships — right now I’m still talking in the abstract. The same goes for what “tokens” and “objects” are (very importantly: these in no way, yet, involve any sense of semantics, representation or content).

³¹ At the same time, the following should not be confused: while each state, transition and total effective procedure is *finite*, the formal definition of the Turing Machine requires an infinite “tape”; this is to allow for finite states that may be *arbitrarily large*.

get their corresponding output state. An *intensional* definition, on the other hand, is a definition of *how* the function is to be computed (this description is also called an *algorithm*). This defines the function as a series of steps. The purpose of presenting these different ways of specifying or defining a function is to highlight the fact that a function is an abstract description of what is done (either step-by-step for intensional definitions, or input/output correlations for extensional) by a computation. This allows us to talk about different computations as computing different functions. And we can characterize different *kinds* of computations according to the different *kinds* of functions they compute.

This basic specification of computation and the terminology that goes along with it is actually the result of the work of several brilliant mathematicians, most notably Alan Turing (1936) and Alonzo Church. This definition is believed by most mathematicians to be the best (most comprehensive) formal, mathematical characterization of the intuitive notion of computation. Because of the elegance of Turing's characterization, the class of functions computed by this formal definition is said to be *Turing-computable*. It is a further claim, however, to say that *all* computations are Turing-computable. This further claim was boldly proposed by Church, and thus bears his name along with Turing's: the *Church-Turing Thesis* — and this is the first of the two premises to the computational hypothesis. Again note that the Church-Turing Thesis would be false if one, or more likely a set, of computations (functions carried out by effective procedures) were discovered which were not Turing-computable.

This last point bears some further discussion as I should characterize which functions are computable and which are not, and this will figure prominently in the ensuing discussion of the plausibility of computationalism as an account of mind. The main distinction that has been made for functions is the difference between digital and continuous computation, and analog processing. These are characterized according to the kinds of *sets* that the computations can be effectively computed over — these sets are often referred to as “number-lines,” to invoke an intuitive sense of how to conceptualize the properties of the sets.³² Three sets are relevant: discrete, dense, and continuous. *Discrete* sets are sets with gaps between their members such that every element of the set has a nearest neighbor on all sides. If given a member of the set, one can always find the *next* member of the set (i.e., the set is *countable*). Discrete sets include the set of natural numbers ($N=\{0,1,2,\dots\}$) and the set of integers ($Z=\{\dots,-2,-1,0,1,2,\dots\}$). *Dense* sets are sets where between any two elements there is always another element. The set of rational numbers (Q , together with all the negative and positive fractions) is a dense set. Here, unlike discrete sets, there is no notion of *the* next member because no matter how small a quantity you add to one member to get a second, there will always be a third between the first and second. But, rationals do still have gaps (namely, irrationals: such as π , $\sqrt{2}$, etc.). *Continuous* sets are sets with absolutely no gaps between elements. The set of real numbers (the set of rationals together with the set of irrationals) is continuous.

Digital computation is defined as computation over a discrete set. This is the classical notion of computation (it should be pretty clear from the above characterization

³² Note that this is also the beginning of a suggestion for a theory of interpretation, as number-lines and sets are also used as epistemological tools in the characterization of physical measurement.

that talking about entities and tokens that are finite in number is well within the boundaries of discrete sets). Digital computation is formally independent of time, although no actual computation implemented (see below) in a physical system can be time-independent. *Continuous computation* is defined as computation over continuous sets. There has been controversy as to whether such computation is indeed equivalent to the kind Turing characterized. However, recent work in mathematics has succeeded in extending the classical discrete notion of computation to that of continuous sets (Blum, Shub, and Smale, 1989). These two kinds of computations are distinguished from *Analog processing*, which is generally taken as *not* involving classical computation (digital) or some extension of it (hence, Dietrich's (1994) characterization of "processing," rather than computation). Analog processing involves a direct, quantitative relation between input and output: as input varies continuously, so do the outputs. It involves operations over continuous sets, but is usually considered as different from continuous computation. It is often described in terms of differential equations. Also, time is often invoked as playing a crucial role as special time-relations within the processing. Analog processing is sometimes also taken to be "implementationally-dependent," meaning that the very identity of a given analog process depends on how it is realized in some physical material so that if you change which material it is instantiated in, you change the process.

We can see that if intelligence turns out to be an analog process, then computationalism may be false (by the second premiss of the computational hypothesis). There is, however, an argument which suggests that even if intelligence were actually an analog process (ignoring for a moment the issue of timing), computationalism would still be able to capture and account for it entirely — at least as far as countering the claim that analog processes capture important functions that take place "between the cracks" missed by computable methods. This is an epistemological argument: what we know about our world ultimately comes down to what we can perceive of it (measure) and the inferences we can make based on such measurements. It follows from this that our knowledge of cognition comes from what we can measure (and the inferences we make based on those measurements). In principle it is possible to make our computations so precise that they can replicate any function to the point of accuracy down to the level of thermal noise and the Heisenberg Uncertainty Principle. In other words, we would be so accurate that you couldn't measurably tell the difference between an actual analog function, and our digital replication of it. Therefore, the claim that intelligence is an analog process (suspending the concern about time) and thus not possibly instantiated as a computation implies that what is important about intelligence is something we can't measure — which seems absurd. So, at least as far as the issue is over precision, computationalism is sufficient.³³

Note, however, that if it turns out that one of the necessary conditions of accounting

³³ A possible rejoinder is that analog processing may capture the complexity of state changes more parsimoniously than computational descriptions. A flaw in this line of reasoning is that it is based on a particular conception of computation, probably derived from a von Neumann-style linear architecture for computation; this, however, is just one kind of computation. Many other styles of computation exist (e.g., parallel distributed processing and connectionism are examples of these alternatives; even these may be simulated on digital von Neumann machines), and this argument may then reduce to an argument over what *kind* of computation is most parsimonious. So again, from the outset, analog processing does not a priori hold a precision capacity over computation.

for intelligence involves accounting for the specific timing of events (not just a matter of sequence), then computationalism is given a significant challenge. In this case, at the very least, computationalism would have to rely on some timing mechanism to calibrate the system so that all further computation would occur in sequence — such a timing mechanism is, by definition non-computational. Furthermore, a single updating timing mechanism would probably not be sufficient because the timing of each computational step may be important. (It is also not enough to argue that “all instantiated computations are in fact time-dependent and therefore timing is not a problem” as the issue is whether timing additionally *needs to be accounted for* — if it does, then the computational description is not explanatorially sufficient.) I will discuss this issue briefly below, and again in Chapter 4.

Barring the issue of timing, it should be clear that our notion of computation is powerful. Before leaving this section, I wish to define two more mathematical terms that will be useful. The first is a *machine*, which is an abstract mathematical description of a computation (this is another way of referring to the “guts” of a function, or *how* a function is computed — an algorithm; in this case I am referring to the intensional definition of a function). A *universal machine* is a mathematical description of a universal computation: given as input the description of another machine (M1) and some input for *that* machine (I1), the universal machine (UM) will compute the function that the input machine (M1) computes with its input (I1). (That is: give to UM as input [M1+I1]; UM then computes the function M1 with I1 as input). The notion of a *computer* can now be introduced as being a physical instantiation of a universal machine. This notion of a computer is of the same kind as the Von Neumann architectures our personal home and office computers are. (There is a difference between physical computers and Turing-computer universal machines: that of infinite memory. But this is a practical constraint, not a theoretical one, as, again, the formalism of infinite memory is solely to capture *arbitrarily large* input/output specifications.) I now turn to discuss the second premiss of computationalism and argue for its plausibility.

3.3 - The Theses of Computational Sufficiency and Explanation

As mentioned above, the second premise of the computational hypothesis is the claim that all cognitive processes are the computations of some function, and that anything that computes these functions is also cognitive. To consider its truth requires expounding more on what the hypothesis entails. Chalmers (1994, p.2) has done just this by identifying two additional theses that underlie the basic computational hypothesis: (1) the thesis of *computational sufficiency*, which claims that the right kind of computational structure suffices for having some cognitive function, and for the possession of a wide variety of mental properties; and (2) the thesis of *computational explanation*, which claims that computation provides a general framework for the explanation of cognitive processes and behavior. The former is an ontological issue (whether thinking is computation), while the latter is methodological: computationalism is not intended simply as a theory of the nature of cognition; it is also offered as the methodology by which to study cognition (having established an ontology usually provides at least the constraints for a methodology). As computationalists we will couch our theories of cognition in the language of computation.

Before establishment of either thesis can begin, there is some further work to do. As Chalmers (1994, p.2) notes, the conceptual foundations of computationalism have to bridge a gap: the mathematical theory of computation in the abstract is well-understood, but cognitive science and AI ultimately deal with physical systems; for this reason, we need a clear mapping between these physical systems and the abstract theory of computation. This mapping will answer the question: what are the conditions under which a physical system can be consistently interpreted as implementing a given computation? Our criteria for such a mapping are that we need to be able to justifiably say that a system in question in some sense “realizes” the computation at some level of organization, and that the computation “describes” the system. This mapping constitutes a theory of interpretation, and lays a significant part of the groundwork for the establishment of computational explanation.

After providing a theory of interpretation, I will then be able to do two further things: First, I will develop the notion of a supervening machine and show how the theory of interpretation may work as a framework for applying a semantics to the supervening machine to make it a scientific model in good standing. This development then allows me to address a second question: what is the relationship between computation and cognition? In answering this question I will outline the minimal ontological commitments computationalism makes for the status of cognition. This answer not only provides the establishment of the sufficiency of computationalism (for now it will be clear how computationalism can describe cognitive phenomena), but along with the developed notion of an augmented machine there is likewise the establishment of the explanatory power of computationalism (how computation provides a general framework for the explanation of cognitive processes and behavior through the use of augmented machines as scientific models).

3.4 - Theory of Interpretation

The theory of interpretation I will develop here is borrowed from Chalmers’s (1994) theory of implementation. While the basic mapping Chalmers developed is kept intact, a subtle but important distinction must first be made, a distinction which spells the difference between interpretation and implementation. In Chalmers’s account, physical systems *instantiate* computations. That is, computationalism is a *metaphysical* claim about the universe: the universe computes functions, which are the same as certain computational descriptions — they are a necessary, fundamental property of the universe. Chalmers’s theory of *implementation* is a strong claim about the nature of the universe; namely, that there is a 1-1 mapping between certain physical systems and the abstract computational descriptions that are instantiated. Chalmers, however, gives no story about what an implementation is other than a realization, so this 1-1 mapping remains as some sort of idealized isomorphism. Here there is no measurement theory — it’s just a given aspect of the world that it computes.

Instead of this strict metaphysical claim, I will remain consistent with Dietrich’s (1990, 1994) account. In this account, a physical system which follows consistent state-changes may be interpreted as computing some function. Here, computationalism is an *epistemological* claim: it’s how we might usefully describe physical state-changes so that we can come to understand, and eventually explain, the processes involved in producing

kinds of interesting physical system organizations which we've labeled as "intelligent" or "cognitive." Thus, I offer Chalmers's mapping as a theory of *interpretation* — a claim which sets how a physical system can be interpreted as making state changes that are mirrored by the state changes of a computational description. Here, the 1-1 mapping is possible only with an application of a measurement theory.

The key change that this distinction makes is that it keeps our computational models within the realm of epistemological tools — along with the rest of scientific models used as empirical machinery. Chalmers's claim is too strong: it forces computational modeling into necessarily dragging in the claim that the phenomena in question is in a deep sense an instantiation of a computation. This is not to say that theoretical models could not make such a claim — but we do not want our computational models to *necessarily* make this claim. And computationalism doesn't need it either. Leaving the model status itself as essentially epistemological avoids this.

The entailments of this change can be seen in how computationalism might fail: the Chalmers account of computationalism fails if it turns out that physical systems really don't instantiate computations exactly (or rejected on conceptual grounds); the Dietrich account of computationalism, on the other hand, fails if computation just isn't useful for describing the kind of regular behavior of systems that are intelligent that we wish to explain (in other words, computation could be the wrong theoretical language to use).

With a theory of *interpretation*, this sets the foundations of the general way in which a mapping is to be made. The task is then left open to the computational scientist to decide what the criteria are for picking out consistent *kinds* of physical systems and consistent *kinds* of computational machine³⁴ entities will be for a mapping to establish the machine as a scientific model — i.e., to develop a measurement theory.

Furthermore, the Dietrich version of computationalism is still just as powerful as the Chalmers version in the sense that it can still answer the following important questions which Chalmers highlights as important for establishing computationalism's sufficiency and explanatory force:

- (1) What are the conditions under which a physical system implements a given computation?
- (2) What is the relationship between computation and cognition?

With this distinction made, I can now continue with Chalmers's basic outline of the mapping, ratifying it for a theory of interpretation. The fundamental statement of this theory is as follows:

³⁴ In the sense of machine defined above; in a moment, this will be generalized to include supervening machines.

A physical system [may be interpreted as] a given computation when there exists a grouping of physical states of the system into state-types and a one-to-one mapping from formal states of the computation to physical state-types, such that formal states related by an abstract state-transition relation are mapped onto physical state-types related by a corresponding causal state-transition relation.

(Chalmers, 1994, p.3)

(the bracketed changes are mine, following the discussion above.)

In order to cash this out, it is necessary to utilize the above definition of a machine to characterize a kind of machine that I will use as the general computational specification for interpretation. The kind of machine that fits this definition the best is the class of *combinatorial-state-automata* (CSAs). This class is “Turing-equivalent” because a CSA can compute any function that is Turing-computable, and all CSA-computable functions are Turing-computable; in other words, CSAs are just another way of describing what “Turing Machines” can compute.

Chalmers’s choice of CSAs as the formalism to couch the mapping theory is important. CSAs allow us to bypass some unfortunate misunderstandings which have arisen from how to use computational descriptions as models — which is the eventual goal of the discussion in this section. The confusion has surrounded what the “parts” of a “Turing Machine” might correspond to in the physical world. The attempts to find such a mapping constitute a category mistake made regarding how computational states are mapped onto physical states and what explanatory role the level of computational description of these states has. Turing’s original conception of a Turing Machine was that of a machine which goes about performing some task. Previous to Turing’s work, mathematical formalisms, such as the propositional calculus, did not provide a way of formalizing the notion of a *process* — that is, there was no formal way of specifying the steps in an algorithm (an effective procedure). The existing formalisms of the time were essentially declarative formalisms. To arrive at a context-free notion of computability, Turing had to come up with a formalism which captured the notion of a process. Hendriks-Jansen (1996) suggests that Turing’s simple and informative formalization of a process (which he described in terms of a mathematician working out a function on paper) was inspired by an analogy with an assembly-line-like process, in which a machine performs finite operations on “symbols” on an arbitrarily large “tape” (like completely rule-governed operations performed by workers on “products” on an assembly line).

This explanation of computation in terms of an assembly-line-like process was never intended as a literal description of how computations might be physically instantiated (in human brains or other physical systems): Hendriks-Jansen (1996, p.95) notes, “...it should be stressed that Turing did not intend his machine to be a *model* of a mathematician’s thought processes” [my emphasis]. That is, the Turing formalism is just that: a formalism. Taking computationalism as only providing a straight-forward analogy from Turing Machine state descriptions to the system to be explained is clearly the wrong level at which to make use of computational descriptions for explanatory purposes. (Even Hendriks-Jansen appears to take computationalism as only being able to be used as

an explanatory analogy straight from Turing machines to explanandum.) Making the Turing Machine-to-explanandum analogy *could* be proposed as a model in the sense I outlined above (Section 2.1.2), but that would be to constrain the computational description to only being of a small class of interpretable computations: namely, those computational descriptions which have state-changes that can be categorized into the constituent parts that were part of Turing's description of his machine (tape, finite alphabet, internal machine states, read/write head, etc.). Adding these constraints to computationalism in general amounts to adding constraints on the kind of interpretations that can be proposed, so only a limited number of actual systems in the world could be described as computational.

Computationalism does not need to be, and should not be taken as, adding any more constraints to the theory of interpretation than I will outline below. For computational descriptions to be used as scientific models in the way I propose, the computational description must be taken into consideration with a particular theory of interpretation which specifies how the computation maps onto a physical system, just as a theoretical model includes a background theory of what kinds of entities and processes in the model correspond to what kinds in the world. Only with this background theory of interpretation can a computational description come to be *used as a model for kinds of physical process*. In this way, the computational description becomes an exemplar of kinds of state changes that are proposed to be responsible in nature for kinds of observed behavioral phenomena. CSAs help us avoid the mistake of requiring the specific parts of Turing's description to map onto physical states because CSAs are described in generic terms of state-changes, and leave open the kinds of states and state-changes the computation might have. With the CSA description, the mappings only have to be concerned with mapping computational states and state-changes to physical states and state-changes. This frees the researcher to describe any kind of physical state changes that might behave regularly. This is exactly what the theory of interpretation provides: a framework for applying a scientific model semantics — i.e., for providing a specification of a mapping between computational states and physical states for a model — and leaves open what kinds of entities and processes the model could have.

Chalmers shows how CSAs fit the above interpretation description by first specifying how a subset of CSA machines, *simple finite-state automata* (FSAs), fit the interpretation, and then making the definition more robust to include CSAs (also, FSAs will sound a lot like the above development I gave of Turing-computability, making the transition smoother).

FSAs are defined as a set of input states I_1, \dots, I_k , a set of internal states S_1, \dots, S_m , and a set of output states O_1, \dots, O_n . There is also a set of state-transition relations of the form $(S, I) \rightarrow (S', O')$ for each pair (S, I) of internal states and input states, where S' and O' are an internal state and the input at a given time (S' is the new internal state, and O' is the output produced at that time). The conditions for the interpretation of an FSA are then:

A physical system P is interpreted as an FSA M if there is a mapping f that maps internal states of P to internal states of M , inputs to P to input states of M , and outputs of P to output states of M , such that: for every state-transition relation $(S,I) \rightarrow (S',O')$ of M , the following condition holds: if P is in an internal state s and receiving input i where $f(s)=S$ and $f(i)=I$, this reliably causes it to enter internal state s' and produce output o' such that $f(s')=S'$ and $f(o')=O'$.

(Chalmers, 1994, p.4)

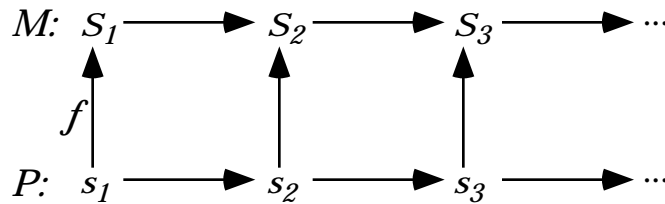


Figure 2.3 - Mapping Physical To Machine State-Changes

Figure 2.3 diagrams the state changes in the physical system P , mapped by function f onto state changes in machine M . This definition, as Chalmers notes, relies on maximally specific physical states s rather than the grouped state-types used in the first interpretation form above. The power of this definition is that it ensures that the formal state-transitional structure of the computation mirrors the causal state-transitional structure of the physical system — this is the key to the interpretation theory.

The use of these simple FSAs has been good to show how a simplified form of my Turing-computation specification above maps easily into this interpretation form. But, the class of FSAs is weaker than the Turing-computable class. For that reason, I need to move to CSAs to finish the interpretation conditions.

The only difference between CSAs and FSAs is that what counted as a single state S in an FSA will now be characterized as an input vector $[S^1, S^2, S^3, \dots]$. This increases our computational power because, rather than accounting for each individual symbol on a Turing machine memory tape, I can now use the vector to specify the components of an overall state, such as all the tape-squares in a Turing machine. There are then a finite number of possible values S_j^i for each element S_i , where S_j^i is the j th possible value for the i th element. This can be likewise done for the inputs and outputs: the input vector is $[I^1, \dots, I^k]$, and the output vector is $[O^1, \dots, O^l]$. State-transition rules are determined by specifying, for each element of the state-vector, a function by which its new state depends on the old overall state-vector and input-vector, and the same for each element of the output-vector. Each element of the CSA corresponds to a distinct physical region within the system (Chalmers notes that there could be alternative ways of specifying this, although it seems certain that whichever way you did would have to be physically possible; Chalmers, 1994, p.5). The new version of the mapping conditions are as follows:

A physical system P is interpreted as a CSA M if there is a vectorization of internal states of P into components $[s^1, s^2, \dots]$, and a mapping J from the substates s_j into corresponding substates S_j of M , along with similar vectorizations and mappings for inputs and outputs, such that for every state-transition rule $([I^1, \dots, I^k], [S^1, S^2, \dots] \rightarrow [S'^1, S'^2, \dots], [O^1, \dots, O^l])$ of M : if P is in internal state $[s^1, s^2, \dots]$ and receiving input $[i^1, \dots, i^n]$ which map to formal state and input $[S^1, S^2, \dots]$ and $[I^1, \dots, I^k]$ respectively, this reliably causes it to enter an internal state and produce outputs that map to $[S'^1, S'^2, \dots]$ and $[O^1, \dots, O^l]$ respectively.

(Chalmers, 1994, p.4)

Chalmers goes on in his original account to discuss the merits of this scheme, as well as its explanatory value. For my purposes here, however, this is enough to show how we can robustly describe a physical system as being interpreted as some computation. This mapping allows me to specify kinds of causal organization via computational description.

Several important issues must be addressed at this point. First, there is the question as to whether this mapping scheme is arbitrary, reducing it to vacuousness. For example, can every system be interpreted as some computation? According to this mapping scheme, yes. But this does not make it vacuous — it would only be vacuous if every system could be interpreted as *every* computation. Can every system be interpreted as *any* given computation? There is some debate as to the answer. Chalmers argues that the answer is “no” because the added requirement that the mapped states must satisfy reliable state-translations rules makes the chances of arbitrary computation-application very slim. I argue that, while it may be logically possible to come up with some mapping to fit any given computation, this is constrained by natural laws (such as physics) and, in combination with the reliable state-translation rules, this greatly constrains the possible mappings.

A third attempt at the “vacuousness of interpreting computation” claim is: if even physical processes like digestion are computations, does that make computationalism vacuous? As Chalmers (1994, p.8) points out, this commits a category mistake with respect to the defining property of digestion, contrary to what computationalism argues is the defining property of cognition. Simply put, instances of digestion will implement some computation, but a system interpreted as computing this computation is in general irrelevant to its being an instance of digestion. By contrast, computationalism argues that it is *in virtue of* being interpreted as some computation that a system is cognitive. In other words, computationalism claims that there is a class of computations such that any systems that can be consistently interpreted as computing them would be considered to be cognitive. (Note that this class has not yet been specified; so this is not to be taken as claiming that just because it can consistently be interpreted as any kind of computation it is therefore cognitive — cognition is a specific class of computations). This claim is also what distinguishes computationalism *as* a kind of functionalism (I will return to this point, below).

3.5 - Supervening Machines: computational descriptions become models

Another question to be addressed is whether a given system may be interpreted as instantiating more than one computation. The answer to this is an emphatic *yes* — in

fact, this is part of the great power of this theory of interpretation: the ability to account for *supervening machines* (SVMs). First, a note on terminology: while the term “virtual machine” is commonly used to refer to the class of possible descriptions I am after, I believe it is an unfortunate term because it suggests that the machines being referred to are somehow “unreal.” Contrast with the usage of *virtual* in modal logic: “virtual worlds” — these are not actual worlds but possible worlds; they are states of affairs that might exist but don’t currently. SVMs, however, *do* currently exist and are quite real (even if they are being used as a model exemplar of how other systems in the world *might* work). I therefore use the term *supervening*³⁵ in place of ‘virtual’ because it more accurately captures the notion that these are functional descriptions of complex systems which have multiple levels of descriptions.

In such systems, the functional description of the system’s organization could at a low level be seen as a certain computational machine, but could also be described at a higher level at which a different type of machine operates (with different computational properties, although these properties depend on the lower-level machine’s operation). In this sense, the “lower-level machine” can be viewed as being “augmented” to produce the functional organization of the “higher-level machine”; the higher-level machine thus “supervenes” on the lower-level behavior of the system. The purpose of this section is to further explain this supervenient relationship, and show how SVMs can be used as computational scientific models in good standing.

There are a variety of different notions of what an SVM is. Clark (1993, p.151) defines a “virtual machine” with respect to the “surface” or external, public behavior of some system. He uses this notion to explain how connectionist models can account for developmental stages in which a system changes from one kind of behavior to another, while keeping the same basic underlying physical architecture. The notion of change in kinds of behavior is important because it introduces the idea of *kinds* of functions being computed, which is certainly an important part of defining the kind of machine (or layers of machines) responsible for that behavior.

However, while Clark has recognized the important role such descriptions can play in the explanation of a system’s behavior, I believe that the internal state changes of the machine are *also* important for explanatory purposes — not just the input/output correlations that make up the observed behavior. Clark’s definition is too weak. That is, while the basic underlying kind of architecture of the system may remain the same (e.g., in a standard connectionist system, the same topology of connectivity between nodes and direction of activation-flow remains the same), the functional organization may be changed radically according to how it processes information, responds to input, etc. (in the connectionist system, the topology of the weight-space for the connections between

³⁵ I am borrowing the terminology of *supervenience* from the technical notion introduced by Davidson (1970) and Kim (1978, 1993). Their more technical usage rests on the definition of *logical supervenience*, which states that, “B facts/properties logically supervene on A facts if no two logically possible worlds are identical with respect to their A facts while differing in their B facts.” I do not intend my discussion to wander into modal considerations and possible world semantics; I am only seeking to use the idea that in complex systems which have multiple levels of description, if you fix the lower-level facts, then you automatically fix all the other facts above, while at the same time, some properties may only exist at higher levels of description (i.e., they are not found at lower levels).

nodes may change dramatically — this is change in the *functional* architecture of the network). These latter descriptions of how the system processes information or responds to input may also constitute kinds of SVMs. Even in connectionist SVMs, there are internal kinds of processing that should correspond to kinds of computational processes in nature; it's just that these SVMs are defined in terms of kinds of vector state-changes over periods of processing — presumably the SVMs found in nature are also defined according to similar kinds of vector state-changes.

An SVM is therefore a *machine* as defined above (at the end of Section 3.2), but includes reference made to kinds of internal state-changes which lead to the computation of the function (i.e., an intensional definition of the function computed). In keeping with the theory of interpretation and the role SVMs are intended to play as scientific models, SVMs are *epistemological entities*. As models, they are intended as computational interpretations of the processes that exist at a certain level of description of a physical system; the SVM can then serve as a public demonstration of how the processes in the physical system work (even if such actual processes aren't directly open to the public), provided the SVM model is a good one. Why call these SVMs separate machines even though they may be instantiated on other systems, or there may even be several SVMs in the same system at the same time? Because the relevant function(s) to be explained by each SVM is different.

Before I can arrive at a satisfactory definition of SVMs as computational models, I must go into more detail concerning the relations between the levels of description which SVMs describe. I start by making clear through some examples what the nature of the identity of entities picked out at a given level of description is, relative to other levels. This will serve to illustrate two important aspects of these levels: (1) the dependence of higher-level properties on lower-level properties, and (2) the possibility of multiple realization. This leads to the notion of contextual autonomy of SVMs, which, in turn, completes the definition of SVMs as computational models.

The type-identity of the kinds of internal state-changes that are part of an SVM, and how they are implemented in types of physical states, requires some explanation. Ron McClamrock (1995) treats this issue nicely in his discussion of the contextual autonomy of levels of kinds of properties in complex systems. As he points out, it is generally accepted that complex systems may have multiple levels of organization. These multiple levels are standardly viewed as roughly hierarchical, with the components at each ascending level being some kind of composite made up of the entities present at the next level down. Explanations may then be provided for the system's behavior at higher (coarser-grained) and lower (finer-grained) levels. Thus, there may be levels of SVMs, corresponding to levels of organization that are responsible for kinds of behavior.

McClamrock gives a couple of examples of complex systems with levels of behavior, including that of an organism and that of a computer. An organism may be explained at various levels of organization, including (but not restricted to) ones that are biochemical, cellular, and psychological. A similar treatment of levels can be given for the physical system which constitutes a standard personal computer — I'll give this one in more detail because it is more familiar and because current computational models are generally

instantiated on such computers³⁶ (although similarly detailed stories can now be told in other sciences involving complex systems, such as the multiple levels found in biology).

At the “hardware level,” there is consideration of the architecture of the computer according to physical components of the system we might identify if we were to open the computer chassis up. For example, my PowerMac 7100 has a PowerPC 601 CPU, a bus (for communication amongst other hardware components of the computer architecture and external input/output devices), a power transformer, RAM memory, and other computer hardware. These components are organized to allow electrical charges to travel through and be held in the computer in complex arrangements.

It is important to note here that it is misleading to think that there is somehow a “bottom-level” description which is ultimately all there is in the sense that every level above is “just” an abstraction. While it is true that there is only one physical universe, a part of which is being described at a given level of organization, each subsequent “level” of description is just another way of categorizing the physical world (even my description at the “hardware level” of the parts of my PowerMac are one such level). None of these levels of description is *a priori* privileged as “purely objective” (although it may be the case that the entities at a “lower level” may have more common instances in other parts of the world than entities at a “higher level” of description, and we may be more certain about our description of one level over another). Thus, despite what level of functional description we might make of my PowerMac 7100 (anywhere from the lower “hardware level” to the higher “software” levels), they are all part of the same complex physical system.

All of the components of the computer architecture can be described at differing levels of “granularity” (coarser or finer) — and each of the entities denoted by the description exist only in the context of the system being organized appropriately. For example, at the “machine-hardware level,” some of the descriptive entities include the machine registers in the CPU, logic gates which determine “flow” of electrical charges through the CPU architecture, and memory locations in the RAM which hold charges which may play a role in further system processing or be modified. A level “up” (a coarser grained view of the system) might be the level of the “operating system.” Here the descriptive entities are now data structures (such as stacks, lists, etc.), control flow operations (e.g., hold a value in memory, read the value at some memory address, or wait for input from the keyboard), and basic logical operations. At a level “above” the operating system we might then consider running programs, such as the Macintosh Finder, or the WordPerfect word processor (and there might be several levels of description in between).

³⁶ It is important to note here that current computational theory describes potential computations that could be instantiated in other architectures besides the basic von Neumann-style serial processor architecture that most of our current personal computers are based on. SVMs may be implemented on parallel computational devices — computations in an SVM might be carried out simultaneously in the computation of the function (part of the reason why the CSA formalism is couched in terms of state-vector transitions instead of single-state transitions). Thus, it is again important to note that we should be wary of getting trapped in the practice of only considering the architecture that current computational devices have as stipulating the capacity of computation in general.

As you go up each level, the current level depends on the organization of the entities at each level of description below it for its existence (this draws on the notion of dependence found with logical supervenience, which is why I use this term). For example, if at the hardware level the system was not able to maintain consistent machine registers, then functioning at the operating system level or “higher” software levels would not be possible — many of the descriptive entities at these higher levels might not have the requisite conditions to exist. Failure to operate at a higher level, like the operating system level, however, does not necessarily mean that the basic entities at the machine level can’t still exist. On the other hand, each higher level has new computational properties which only exist in the context of that level — not below — and these entities may serve to constrain the overall organization of lower-level entities. For example, at the level of a working word processor, it is relatively simple to carry out commands like “copying” a set of represented text and “pasting” it in a different location of the document. Certainly this operation could likewise be described at the machine level (after all, the operation at each higher level entails that the lower levels instantiate the operation), but would require a much more complicated a description — describing text manipulation at the level of a functioning word processor is much more parsimonious: provided that the lower-levels have been organized appropriately, the “copying” and “pasting” commands may be accessed by the machine running at the “word processing” level by a simple procedure call.

Another descriptive term must be introduced to further characterize the complex relation between levels: *multiple realization*. The basic notion of multiple realization is that one and the same descriptive entity at a higher-level might be consistently realized (instantiated) by a number of different lower-level system organizations of lower-level descriptive entities. McClamrock (1995, p.24) gives a good example of multiple realization:

“A particular case of multiple realization and higher-level generalization within a system worth noting is the implementation of higher-level primitives in computer programs, as it provides a particularly clear example of the possibilities of multiple realizability even *within* a particular complex system. Consider a particular variable in a program. At different moments in the running of the program, the machine-level implementation of the variable may be quite different. It will, of course, hold different values at different times; but furthermore, it will also reside at different real memory locations (as garbage collection may re-allocate variable memory) and so on... What makes all those implementations of that variable an important and interesting class of things is that they play a fundamental and reliable role in the processes of the system at the higher level of organization. They may not share any terribly interesting properties at the machine level that they don’t share with, say, implementations of other variables used in the system. And of course, as always, the way in which those variables are implemented on *different* machines will typically be even *less* likely to have anything other than their higher-level functional properties in common.” [See Wilson, 1991, for the original description of garbage collection of variables.]

Thus, returning again to the example of the parsimonious description of text manipulation, the lower-level description of machine hardware might be complicated further in that there might not be any consistent “location” of the given text in real memory, but it could instead be kept track of only on a coarser-level view of the machine operation (kept track of by the word processor SVM).

Given the potential of multiple realization of entities at higher levels by different, but nonetheless appropriately organized lower-level entities, it should be clear that SVMs may be implemented in such a way that they might share some of the same computationally instantiated states as other computations for other machines — that is, a system may implement more than one computation, and at a variety of different levels, or even a variety of different computations on the “same” level. Again, a classical example of this is to consider a reasonably complex operating system on a computer: right now there are multiple SVMs running in my computer, such as WordPerfect, the Macintosh Finder, the Macintosh Operating System, and all the various component machines that might make up or be used by these machines. And they are all using the same physical architecture, and also all have consistent mappings that map the causal organization of the system onto the abstract machine descriptions of the computations being performed.

A complex relationship thus exists between the levels of operation of a system. On the one hand, the operation at a higher level of a system depends on the operation at the lower level of that system: the lower level must instantiate the kinds of entities and processes which the higher level depends on. On the other hand, the kinds of entities which the higher level requires might have multiple kinds of realization, so that a *particular* kind of lower level may not be necessary — there may be a class of different lower-level systems which are sufficient for instantiating higher-level systems. And for my purposes, what I am interested in is the function (its intensional description) being computed at a particular level (or levels) of SVMs. This explains the *contextual autonomy* of an SVM: it is autonomous because it may be realized in a number of different types of systems, but it is contextualized in that there are constraints on what kinds of systems there may be which can succeed in realizing the resources³⁷ required by the SVM.

With the above explanations, I can now define *computational models* as the SVMs which are picked out by reference made in an accompanying theory (including the measuring theory) to a particular level of description in a physical system, and what physical state-changes in the world the SVM state changes correspond to. It is these SVMs with accompanying theoretical claims that may serve as a models. As mentioned above, it is not necessary, and may even be inappropriate, to make reference to the underlying computation and data structures involved in instantiating the SVM in the computer it happens to run on when making the comparison between it and the physical system which it is intended to model. In fact, *at* the level of the description of the SVM, what behaviors or processes exist in the SVM models need not be couched in terms of data structures and control flow operations at all. What is important for the SVM’s explanatory power is that its state changes *at its level of description* are shown to

³⁷ These resources are often considered in terms of “computational resources”: what lower-level functions need to be computed — and these functions are often times only considered in terms of their extensional (input-to-output pairing) descriptions.

correspond to theoretical entities and state-changes in the world — that is, we can show that the SVM's entities and state-changes correspond to an underlying ontology common to the SVM's description and the phenomena to be explained.

At the same time, the particular level of description of the SVM is also parsimonious in the sense that, with respect to modeling other systems (namely, those naturally occurring), the SVM model may be isomorphic with respect to its computational description to the system to be modelled, even though its underlying implementational levels are different. That is, while the SVM on a computer has lower levels of machine hardware which instantiate it, in contrast to the modelled system, which may be instantiated biologically, they may still be isomorphic at the stipulated level of description³⁸ because of the property of multiple realization — the underlying differences in types of systems is irrelevant as long as the same kinds of computational properties are sustained at the compared levels of description.

The SVM model thus serves to partially establish Chalmers's first thesis: the thesis of computational explanation — computationalism uses SVM models, which provide a general framework for the explanation of cognitive processes and behavior. I say that this has only been partially established because it still needs to be shown how cognitive processes can, in fact, be captured by SVM descriptions. Completion of this thesis, as well as the thesis of computational sufficiency, come together in Section 4, which I turn to now.

4 - Computationalism's minimal ontological commitments to cognition

Now that I have presented the theory of interpretation (which explains how a physical system can be consistently interpreted as implementing a given computation), and how an SVM (a computational description of a descriptive level in a complex system) can be part of a scientific model in good standing, I turn to the issue of presenting computationalism's minimal ontological commitments for cognition. This will entail completing the establishment of the thesis of computationalism's sufficiency (that the right kind of computational structure suffices for having some cognitive function and, subsequently, the possibility of possession of a wide variety of mental properties), and in turn, complete the establishment of the thesis of computational explanation (that computation provides a general framework for the explanation of cognitive processes and behavior).

As mentioned above in Section 3.3, this establishment is accomplished by answering the following question: What is the relationship between computation and cognition? The answer to this question, as Chalmers points out, lies in the fact that the properties of a physical cognitive system that are relevant to its implementing certain computations are precisely those properties in virtue of which the system possesses mental properties (and therefore a computational description in an SVM could explain the system's cognitive processes) (Chalmers, 1994, p.3).

³⁸ ...the level of the SVM and the level which it is meant to correspond to in the modelled system, described, in part, by the measuring theory...

A central property of computation, along with the theory of interpretation which endows SVMs with the capacity to describe systems at varying levels of description, is that computations can be abstract specifications of the causal organization of physical systems. And this is the connection needed for explanation of cognition; as Chalmers puts it, “Causal organization is the nexus between computation and cognition” (Chalmers, 1994, p.11). To capitalize on this specificatory ability, Chalmers introduces the notion of a *causal topology* of a system, which is the abstract causal organization of the system, conceived of as a pattern of interaction among parts of the system, abstracted away from the make-up of individual parts and from the way the causal connections are implemented (Chalmers, 1994, p.11).

This notion of a pattern of causal interactions is then used to define an *organizational invariant*. A property *P* is an organizational invariant if it is invariant with respect to the causal topology — i.e., any change to the system that preserves the causal topology will also preserve *P*. The changes that may occur for which an organizational invariant must remain intact include: 1) moving the system in space; 2) stretching, distorting, expanding and contracting the system; 3) replacing sufficiently small parts of the system with parts that perform the same local function; 4) replacing the causal links between parts of a system with other links that preserve the same pattern of dependencies; and 5) any other changes that do not alter the pattern of causal interaction among parts of the system.

Here we can now see how the identity of a physical process like digestion (recall the third attempt at arguing for the vacuity of computationalism at the end of Section 3.4, above) really is of a different kind than cognition. If we were to replace all the parts of a digestive system with pieces of teflon, while preserving causal patterns, after a while the system would no longer sustain digestion — it would cease to satisfy the particular identifying features necessary for digestion, such as breaking food down, extracting energy, etc. For cognition, on the other hand, computationalism asserts that the defining features *are* its particular causal topology. This claim is the minimal ontological commitment computationalism makes for the identity of cognitive or mental phenomena. Any additional constraints constitute an additional theory of cognition beyond computationalism.

Now for the connection to cognitive properties. Chalmers, as well as others in the general philosophy of mind camp of *functionalism*, such as Armstrong (1981) and Putnam (1967), have argued that cognitive properties are effectively defined by their role within an overall causal system: it is the pattern of interaction between different states that is definitive of a system’s cognitive properties. The majority of cognitive properties have classically been viewed as depending only on the “internal” states (states within the boundaries of the brain). These could be contrasted with cognitive states which are dependent on the context of the environment (e.g., some definitions of knowledge, which require that a system knows *P* only if it is in an environment in which *P* is true). The “internal” states clearly may be computationally described because of the notion of internal organizational invariants described above, and therefore pose no problem to computationalism — the idea of a causal topology contributes as much as the brain causally contributes to cognition, and an SVM could describe the causal topology. In other words, as long as the cognitive function to be explained is a matter of the causal processes going on inside the brain, an SVM could be created to describe the function.

that constitutes those processes.³⁹

Recent development of positions in cognitive science and AI (spearheaded by the situated cognition movement), however, have argued that many of the cognitive phenomena considered (even those that have classically been treated as “internal” subjects, such as memory, planning, reasoning and representation) are actually a product of complex interactions with a structured environment. Because of the important role of the environmental context, along with the history of interaction which leads to the development and expression of these cognitive phenomena, it is argued that their full explanation requires an account of not only internal system organization, but also the structure of the environment, and how interaction between the system and that structure in the environment lead to internal changes over time (in the context of a history of interaction). The potential challenge, then, is how a traditionally “internally-focused” cognitive science might handle computational accounts of these phenomena.

This, in fact, is not a problem for the SVM descriptive framework I’ve outlined above. The computational model that would be required to model interactive phenomena might be construed in a variety of ways. One would be to espouse a version of “wide computationalism,” in which the cognitive function being computed is “extended” beyond the cognitive agent to include environmental functions. Here, an SVM modeling the situation would have constituent parts, including the agent modelled *and* the environment, and the interactions which take place between them. The function which the SVM as a whole computes, however, would constitute the cognitive function distributed over the “boundaries” of the agent into the environment.

Another approach would be to have several SVMs which might interact with each other: one of the SVMs could be the agent, designating the internal functioning of the agent and how it reacts to stimuli and also how that internal functioning might change over time based on its interactive history. That agent SVM might then interact with another SVM, which constitutes the environment. The environment SVM would govern how the environment behaves and presents itself to the SVM-cognitive-agent. This style of modeling might be preferable in cases where it is important to designate strict “boundaries” as to where the cognitive agent ends and the environment begins. Nonetheless, such compound models would capture the situated cognition requirement of including the environment and interactions in explanation. (The environment might also be divided into a number of SVMs, each describing active objects in the world.) (Bickhard, 1980a, in fact proposes such a model.)

A third kind of approach to modeling might be a radical form of modularity, in which the cognitive agent is considered to be a collection of interacting individual SVMs, each computing their own modular cognitive function — although this collection could in most cases just as well be instantiated as a single SVM function that is nomically split into separate sub-functions.

These kinds of approaches to modeling are becoming increasingly numerous, as found in the growing number of artificial life models (e.g., Agre & Chapman, 1987; Beer, 1990; Drescher, 1991; Lindgren & Nordahl, 1994; Ray, 1994; Sims, 1995; and Terzopoulos, Tu & Grzeszczuk, 1995), and their explanatory power is likewise becoming

³⁹ Again, for the moment I am suspending concern over issues of capturing timing relationships between functional components in the causal topology.

more evident. And none of these modeling techniques is outside of the reach of computationalism's SVM modeling framework.

This, however, is not to say that there aren't some kinds of phenomena which computational models as SVMs don't capture — namely, those that are patterns of causal interaction within a system, but aren't captured by computational descriptions. I will not go into detail here concerning arguments for whether these are important to cognition, or how they might be avoided. Rather, I will just note several of the major possible threats. As already mentioned above, if it is demonstrated that certain cognitive phenomena are produced by causal interactions which are necessarily analog in nature, then an SVM model could not possibly capture the relevant behavior to be explained. Another issue is that of timing: computational descriptions make no formal account of timing, and were cognitive phenomena shown to be necessarily timing-dependent “throughout the expression of the phenomena in the system” (i.e., it's not just a matter of the order of steps in the procedure, or a matter of calibrating the SVM to one or a few non-computational timing devices, but instead requires specific, independent timing throughout the functional activity of the system) then an alternate framework would have to be found.

Another issue, which on inspection turns out to not be a threat, is that of potential “phenomenological” properties of a physical system. A certain degree of skepticism about the definition of cognitive states according to causal invariants can be relieved by the following important distinction between two kinds of mental properties: psychological properties (which are characterized by their causal role; including beliefs or concepts, learning, and perception), and phenomenal properties (which are characterized by the way they are consciously experienced). This is an important wrinkle to the story of cognition as causal invariants because phenomenal properties, or phenomenological consciousness, are often mistaken as having a causal role in the determination of behavior. Chalmers (1994, 1996) argues, via a reductio, that phenomenal properties can play no causal role. The argument is, roughly, that you couldn't have a system with a particular causal topology even “notice” a change in its conscious experience without changing a bit of its causal topology to allow it to react, as “noticing” would require (even internally so as to entertain a “thought” that it noticed something) (Chalmers, 1994, p.13). Therefore, while phenomenal properties may still be mysterious, it should be clear that a causal topology captures everything else of interest to cognitive science. (See Chalmers, 1996 for an extensive elaboration of what these phenomenal properties might be, and what a science of consciousness might look like.)

Now for the final move to establish the sufficiency thesis, and subsequently establish the explanatory thesis. Any organizationally invariant property will depend only on some pattern of causal interaction between parts of the system. And, given the above theory of interpretation, any such pattern can be straightforwardly abstracted into an SVM description, as the parts of the system will correspond to elements of the CSA state-vector, and the patterns of interaction will be expressed in the state-transition rules. Assuming that Chalmers's reasoning is valid, this should establish the thesis of computational sufficiency, because, assuming that the cognitive phenomena we are interested in are, in fact, constituted as patterns of interaction between different states of the system, then an SVM can describe such a causal topology. And this, in turn, establishes the rest of the thesis of computational explanation, because computational

descriptions in SVMs can effectively be used to explain the cognitive phenomena they model (which, according to the thesis of computational sufficiency, potentially includes all cognitive phenomena).

An important final note to make before giving the detailed description of the handcoding critique with respect to computational cognitive modeling concerns computationalism and the issues of semantics and intentionality. The characterization of computationalism and its theory of interpretation, as I have presented it, makes no account of potential cognitive semantics (“meaning”) or intentionality (the property of a system or state of a system which is somehow “about” something else — a central property of representation).⁴⁰ Chalmers (1994) and other computationalists properly identify this as one of the important *strengths* of computationalism: neutrality with respect to semantics and intentionality. Accounting for semantic or intentional content is a separate issue (I will deal with this issue in great detail in the following chapters). Some have inferred from this that computationalism misses an account of cognition entirely because it does not account for semantic content. But this would only be so were it to turn out that an account of semantic or intentional content was not found in a causal topology. I argue that such an account can be made using the framework of computationalism, while not having to build such an account into the definition of minimal empirical computationalism.

5 - Framework for the handcoding critique as a methodological tool in computational cognitive modeling

At the end of Chapter 1, I concluded that the identification of (and proposed attempts to avoid) particular kinds of handcoding is integrally dependent on the contrast between current competing theories funding the particular model(s) in question and the positions that are reacting against those models. In order to answer the question of when handcoding is legitimate or not, I had to first define the explanatory framework for which I will be using the handcoding critique: computational cognitive modeling — particularly aimed at developing the technical vocabulary for SVMs). Now that I have done this, I can proceed to outline the roles that background theories play in the determination of what is handcoded legitimately or illegitimately, and therefore how the framework of the handcoding critique is to be used.

The framework for the handcoding critique plays two roles in the logic of explanation. First, it determines *where* in the logic of a model’s role in explanation handcoding can cause problems. I have pointed out where much of this can occur already, in the course explaining the use of scientific models in general (in Section 2 of this Chapter). Then, against the backdrop of what the rival background theories take to be the nature of the phenomena to be explained, the framework is used to uncover *what* it is that has been handcoded, and therefore what remains to be explained. With these two

⁴⁰ This is not to be confused with a *model semantics*, which concerns what the model as an explanatory device is about — that is, what in nature the model is intended to pick out and describe. The model semantics is established, in part, by the theory of interpretation, measuring theories and other background theories.

pieces in place, the framework then provides the framing criteria for what needs to be done (and at what level of explanation) to avoid the identified handcoding. This is how the framework for the handcoding critique serves as a methodological tool: to diagnose handcoding, determine what effects it has on current explanation, and to motivate future models which eliminate handcoding so that a deeper explanation can be reached.

I will now briefly summarize what I have shown in the overview of computational cognitive models. Sections 2 through 4 of this Chapter have shown that computational cognitive models, just like any scientific model, play a crucial role in the explanation of phenomena, the underlying mechanisms of which we may not be able to directly observe (at least, not currently). Again, the goal of an explanation is to bring about an understanding of such possible underlying mechanisms, where previously there was a lack of understanding. Model exemplars⁴¹ accomplish this by presenting us with a publicly observable “picture” of how such mechanisms might work. Cognitive phenomena, according to computational cognitive science, are construed as a product of the causal topological organization of a physical system interacting with the world. SVM models, along with their background theories, are used by computational cognitive science to describe these causal topological organizations. Among the background theories is that of the measuring theory, which establishes the semantics of the SVM model — this theory picks out the aspects of the naturally occurring system that the processes, entities and behaviors of the SVM are intended to correspond to, and in doing so, attaches these correspondences to a developing ontology of the phenomena being explained. The role of a computational cognitive model in an explanation is to therefore show or demonstrate how a cognitive phenomenon is produced according to the operation of or the functional system organization found in the SVM model — and these mechanisms and processes within the model exemplar are open to public observation.

All models are handcoded to some extent. Aspects of the model that are handcoded indicate what it is that the model does *not* explain. What we therefore want in a good computational cognitive model is a model that gives us a particular picture of the system in which we can discern with clarity the principle(s) of functional organization of the system that gives it the capacity to manifest the particular phenomena we are investigating — that is, we want a model in which what *is* handcoded is *not* crucial to showing the desired functional organization principle for the phenomena we wish to understand. Such a model would be handcoded appropriately. Precisely pinning down what counts as success at capturing all that is important to explaining a phenomena in question, versus what is not important, depends upon on the rival theories being bet against one another and their relative explanatory power. While full determination of the success of one theory over another has to ultimately wait until some hindsight is available (i.e., so that it can reasonably be concluded that one theory is better confirmed relative to another), use of the handcoding critique begins by identifying and making explicit such background theories and weighing their relative explanatory force.

The order in which the identifications are made of what the background theories are, and what, precisely, in the model has been handcoded, is not always clearly separable from the perspective of the historical process of identifying handcoding. Often times, the

⁴¹ Again, model exemplars are the actual physical models; and SVMs are just as real as any other physical model exemplar.

competing background theories and all of their ontological commitments are not clear at the outset of the investigation. Rather, an investigation which employs the handcoding critique typically begins by observing that a model proposed as an account of a cognitive phenomenon leaves out some process, entity and/or function which one would expect (or need) to see in a full explanation — and the proposed theory employing the model in an explanation offers no satisfactory computational solution for accounting for what was left out.

As I discussed in Chapter 1, there are two kinds of observed handcoding that I have outlined which should be held distinct: handcoding with respect to direct researcher influence, and handcoding with respect to the interpretation of the model and phenomena to be explained (which typically implicates the measuring theories involved). The two kinds of handcoding that I have distinguished are *not* indicative of two particular mistakes; rather, they are a way of characterizing two kinds of evidence for a mistake having been made, or the circumstances in which the mistake has been imparted to the model (and the model exemplar's interpretation). These two kinds of handcoding parallel the above distinction in the discovery process between the uncovering of background theories and what, in particular, is lacking in a model. That is, handcoding with respect to interpretation is a central part of the identification of what is lacking in a particular background theory, whereas handcoding with respect to the researcher's inappropriate influence on a model constitutes the discovery of what is particularly missing in the model exemplar (and also why that model might still appear to “perform,” even though it is lacking one or more key pieces of the explanatory story).

Since handcoding with respect to researcher involvement tends to be identified first, I will start with its description. I will again use Giere's analysis of models to aid in picturing “where” in the logic of explanation the two kinds of handcoding have their affect. The proposal made by computational cognitive science is that there are “SVMs in nature” which are the instantiated computations that produce the phenomena, and thus what we wish to describe in our models (these “SVMs in nature” are theoretical assumptions, following the assumption that computationalism is correct and that there are therefore naturally occurring system organizations that produce cognitive phenomena and can be interpreted as implementing an SVM). Given this, the logic of the use of SVM models in explanation can then be thought of as the comparison between the SVM model and the SVM in nature; see Figure 2.4. In this figure, the boxes that have full lines are open to public observation (and thus, understood), but the box with the dashed line (in this case, the “naturally occurring SVM”) is not open to public observation (it is to be explained by the SVM model, based on the relative “fit” between predictions and data).

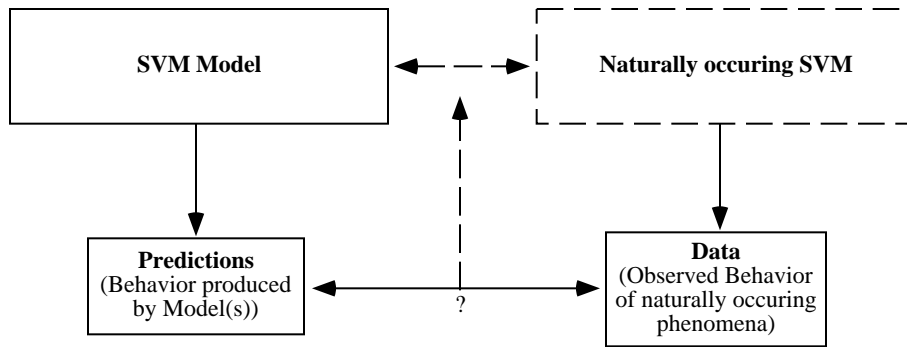


Figure 2.4 - Analysis of SVM model of “naturally occurring SVM”

The logic of handcoding can then be explained as follows. Handcoding with respect to *inappropriate researcher involvement* is a matter of satisfying the following conditional: If a computational cognitive model is only capable of producing the desired phenomena through the involvement of a researcher (who creates the model and may be involved in its run-time operation), such that the production of the phenomena (i.e., the production of what is then interpreted as a prediction) is really a matter of the model *plus* researcher, then the model does not exemplify the whole (or sometimes even any interesting part) of the mechanism in question. Instead, it is the researcher, whose cognitive mechanisms we do not have direct access to, that is responsible (in part, or entirely) for the ability of the model to produce the phenomena. Since the mechanism of the researcher (i.e., what it is in the researcher which allows *them* to produce the necessary function to aid the model) is not open to public observation, we still lack an explanation — that is, we lack a model of the mechanism that produces the phenomena.

Figure 2.5 shows the result of handcoding with respect to inappropriate researcher involvement in the model. The behavior of the model that is being compared to the observations made of the naturally occurring phenomena is really produced by an SVM that is a conglomeration of the proposed model SVM and the researcher’s SVM (the SVM(s) that constitutes researcher’s own cognitive abilities). The position in the analysis of a model’s explanatory logic where the model is intended to reside is therefore pictured within the experimenter SVM, surrounded with a dashed line, indicating that it, like the naturally occurring SVM, is unexplained; only the SVM model within the non-dashed box is open to public observation. The observable SVM model is only providing *part* of the function produced by the SVM model *plus* the Researcher SVM, the total of which is then being interpreted as a prediction. What is damaging to the model’s explanatory power is that the researcher’s SVM is not open to public scrutiny; in fact, the researcher’s SVM is often the very *same* “naturally occurring SVM” that we wish to describe, which leads to a situation that is question-begging: Figure 2.5 might be drawn where the proposed SVM model appears inside the “naturally occurring SVM” box, which makes it blatant that what is to be explained is really being relied upon as part of the proposed explanation of itself.

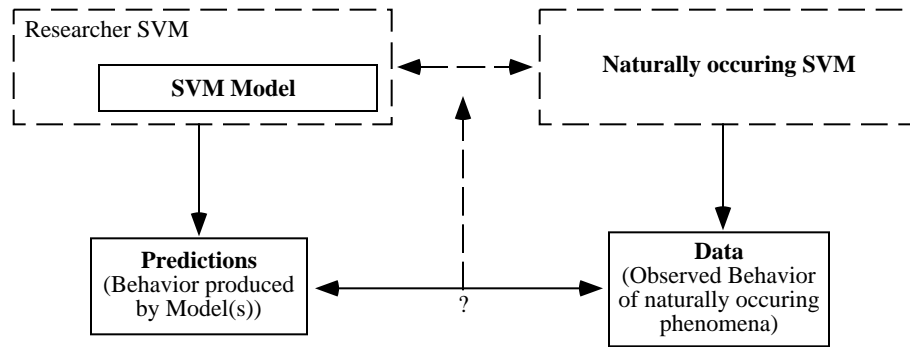


Figure 2.5 - Analysis of handcoding of SVM model

What, in turn, secures the claim that something theoretically important is missing is to uncover *how* the proposed model is a result of *handcoding with respect to interpretation*: the background theory of the model being investigated is shown to have misconstrued the phenomena being investigated; that is, there is an error concerning the fundamental nature — the *identity* — of the phenomena to be explained (or the theoretical entities or processes proposed to play a role in the production of the phenomena). As the model analysis makes explicit, predictions and data, each based on observations of the model and the naturally occurring system (respectively), crucially involve the role of measuring theories: theories which interpret certain phenomena as being of a given kind. While measuring theories clearly relate to what is observed, they also play an important role in establishing the criteria for the identification of what unobservable entities would be. And all of these identifications are directly based on the background theories which propose them. Thus, if the background theories have misconstrued the basic nature of the phenomena, then potential predictions and interpreted data likewise carry the misconstrual.

If the interpretation of these models is in fact false, the measuring theories used for making predictions from the model and interpreting data from observations of the world will be misleading. Such measuring theories may then create the false impression that the proposed models are being corroborated by data collected, when in fact the background theories force the researcher to miscategorize features of the model and/or world. Working within the confines of these mistaken background and measuring theories will additionally cause the researcher to be blind to whole facets of the phenomena that may be central to their existence. For example, as I will discuss in detail in the next chapter, I propose that models of analogical cognition rely on an approach to the nature of representation that does not allow for the emergence of new representational content. Subsequently, an account of such emergence is not possible within these proposed models. Furthermore, representation emergence is rarely mentioned even in theoretical descriptions of analogy-making, although many agree that it happens and make claims to the effect that analogy-making can result in seeing something entirely new in either or both of two compared situations. Putting it bluntly, the emergence that I argue is required of these accounts constitutes an “embarrassment” to these modeling approaches, and they neither talk about it in their models nor “see” it when comparing their models to the world (Camac & Glucksberg, 1983, are one amongst a few rare exceptions).

There is an interesting (and deep) parallel between what I am arguing current analogy modellers are doing and the situation that Mendel faced in his attempts to categorize peas, as described back in the discussion of measuring theories (Section 2.3 of this Chapter). The analogy is between Mendel’s measuring-theory-handcoding and the situation in the approach to modeling the role of representation in analogical cognition. Namely, Mendel *did* discover very important statistical principles regarding the distribution of phenotypic possibilities from particular parents. So, also, analogy research has been quite successful in uncovering important features of analogy-making, including likely requirements for representation structure (again, handcoding does *not* entail an inability to discover *anything* about the phenomenon). At the same time, however, Mendel’s handcoding with respect to interpretation of data led him to interpret variations as unimportant “noise,” when in fact such variations are the direct product of the underlying mechanisms of genetic transmission. Full explanation of this “noise” could not be made until the discovery of DNA and the mechanisms for gene expression. Analogously, I argue that central processes in the production of analogy depend on an underlying model of representation that makes an account of how representational content emerges, is maintained, and can change, and until such an account is given, analogy will only be partially understood at a high level. The detail of this parallel between Mendel’s measuring-theory-handcoding and that of current analogy research will be clear by the end of this dissertation.

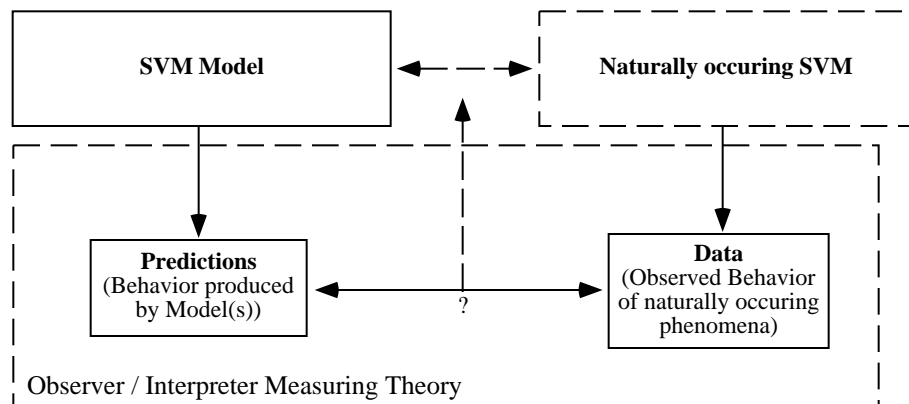


Figure 2.6 - Analysis of handcoding with respect to interpretation

In the model analysis, such handcoding with respect to interpretation would then be pictured as in Figure 2.6. Here, the large dashed box indicating the scope of handcoding with respect to interpretation includes the Predictions and Data, and the processes of the measuring theories which produce them. Clearly, this infects the entire enterprise of the comparison of the model to the phenomena in the world.

To repeat the point made in Section 3.1.3 of Chapter 1, what makes the issue of handcoding non-trivial in cognitive science is that cognitive scientists are studying and attempting to model precisely that which they do naturally: cognition. For many facets of cognition, such as reasoning, language comprehension, representation, perception, and so on, it may be very easy to attribute to the model cognitive properties that it in fact does not itself autonomously manifest. While it’s easy to ground one end of the empirical investigation of the model by considering its behavior — either it produces the behavior

like the phenomena or it doesn't — it is very difficult to ground-out accounts of the nature of phenomena (for example, what representation is). So, while handcoding with respect to interpretation is usually rather a mundane observation once the background theories have been made explicit and it has been exposed how the previous theory misconstrued the phenomena to be explained, in cognitive science the issue of interpretation is very interesting because it is a recognition of the need to remove the researcher's own cognitive and perceptual ability from the models that are intended to ideally model how such cognition comes about (even if it goes against our initial intuitions).

Of course, the ultimate goal is to discharge such handcoding by proposing new models which do not depend on handcoding of the researcher. This is akin to the idea of discharging or removing the "homunculus": the unexplained "little person" that handles the intelligent actions involved. In the case of handcoding, the homunculus in the computational model exemplar is the researcher herself. This is how the framework for the recognition of handcoding can be properly used as a methodological tool: to identify our inappropriate role as unexplained homunculi in our models, and remove ourselves from these inappropriate roles to gain explanatory access to the phenomena we wish to understand.

In Chapter 3, I will explicitly employ this framework to uncover handcoding that exists in current computational models of analogical cognition. I am using this investigation as case study to motivate the adoption and development of a new model of representation. In Chapter 4, I will present this new model and explain how it avoids (i.e., does not require) the handcoding which I expose in Chapter 3.

Chapter 3

Handcoding and Representation Analogical Cognition as a Case Study

1 - Introduction

In the past two chapters I have developed the framework for the handcoding critique. The purpose of this development has been to elaborate on the distinction between appropriate and inappropriate handcoding, and where and what in the logic of explanation such handcoding affects the explanatory power of computational cognitive models. As is now clear, inappropriate handcoding in computational cognitive modelling results in gaps in explanation. In this chapter, I make use of this framework to expose how a currently popular kind of computational modelling approach to the study of analogical cognition involves handcoding of a particular kind; this handcoding ultimately prevents a deeper account of analogical cognition and the involvement of analogical mechanisms in the nature of representational and conceptual development.

My intention here, as mentioned in the introduction, is not to present a developed alternative theory of analogy. Instead, I am using the identification of handcoding in models of analogical cognition as a case study to motivate an alternative approach to the computational modelling of representation. Namely, I am working to create a framework for representation which can provide a foundation for a computational account of the development of structured representation: the *situated representation framework*.

Therefore, this chapter is a pivotal one because it is here that I will present the problem which motivates the development of the situated representation framework: avoiding handcoding of representation in models which involve the creation or change of representations requiring some kind of “structure” (as is the case, I argue, within analogical cognition) entails accounting for the *emergence* and *change* of structured representational content. By emergence and change I mean the following: that prior to emergence, no such structured representational content exists, but afterwards, it does; and prior to representational change, the fundamental components of representational structure have a certain content, but afterwards have a different content.⁴² My critique,

⁴² Note: this does *not* entail that there cannot be *any* version of “weak” nativism, which posits that there exist pre-dispositions and constraints on representations, genetically inherited and present at birth (or emergent constraints which appear during development and are partially determined by genetics); however, it does entail that *novel* representational emergence happens at some point in development, and therefore is a rejection of nativism *in-principle* (i.e., necessarily requiring some fundamental set of representation primitives).

while focussing on models of analogical cognition, generalizes to all models which are based on the same modelling paradigm *and* are aimed at explaining phenomena that require the emergence and change of structured representational content.

Two related points (both raised in the Introduction) should be re-emphasized here. The first regards the level at which my critique is aimed, and therefore how to understand the entailments of this critique for the models of analogical cognition I will be investigating. I am interested in how the features of structured representation, of the kind that these models of analogical cognition depend on, emerge and change — specifically, how it is *possible* for such emergence and change, and how we should describe it computationally. As I mentioned in the introduction, the specifics of what kinds of structure and content emerge, and what processes utilize such structure and are involved in its organization, are additional theoretical claims which go beyond the scope of what I am presenting here. Again, I am working on the foundations for an account of representation development; explanations of development are multi-leveled and I am only focussing on a particular computational problem with respect to structured representation emergence and change.⁴³ Therefore, while I believe that the situated representation framework has implications for accounts of cognitive processes that are representation-based (such as the models of analogical cognition that I am about to discuss), I will not be proposing a new account of these cognitive processes here. In fact, the position I take here is that *Structure Mapping Theory* (SMT), the theory of analogical cognition developed by Dedre Gentner (1983, 1989) and her colleagues, is by and large correct, and that what has been found in its development sets many important goals for the kinds of features of representation that the situated representation framework must be able to account for.

The second, related point is this: although I believe that SMT is essentially correct, I do claim that it misses a crucial aspect of analogical phenomena: that the emergence and change of structured representation can occur *within* and *as a product of* analogical mechanisms. Thus, an account of analogical processes will require the integration of an account of the possibility of representation emergence and change — an account which goes beyond SMT's current scope. I will defend this claim primarily on conceptual grounds, but I also argue (in Section 7) that current data strongly suggest that representation emergence and change is very closely associated with core analogy processing. I strongly believe that such emergence and change is implicated in analogical cognition, and I will use this claim as the opposing theory to initiate the identification of potential handcoding.

The point to be made here is that there are two distinct claims I am making: (1) the claim that analogical cognition involves representation emergence and change; and (2) the deeper claim that structured representation emergence and change occurs in development at *some* point. My second, deeper claim is a claim about how representational emergence and change occurs when considering the development of the organism's representational capacity as a whole; the core thesis of the need for situated representations rests on this being true. The first claim, however, is in regards to the "location" in the architecture of cognitive processes of a complex representing organism

⁴³ Although the solution to this problem nonetheless seems necessary for any robust theory of representation development — and learning and cognitive development more generally.

that such emergence occurs (my claim is that it will happen elsewhere as well).

Certainly, if my second claim is false, and no such emergence is necessary anywhere in development, then my first claim is also false. But, this dependency does not hold in the other direction: even if this first claim turns out to be false — that analogical processes are completely separable from the mechanism for the emergence of novel structured representation and change of fundamental components of structured representation (the position that SMT currently holds, at least in its model exemplars) — the second claim may still hold (defense of this second claim will be given in more detail in Chapter 4).

In addition, I must highlight that falsification of the first claim does not invalidate the basis for my handcoding critique here: the critique will nonetheless serve to show how current models of analogical cognition (and the general class of modelling approaches to representation they belong to) do not offer an account of structured representation emergence and fundamental change, and this will pave the road for one of the conclusions in Chapter 4: that such an account is not even *possible* within the *kind* of approach to representation that these models currently assume. So, while an account of representational content emergence and change may ultimately turn out to not be crucial to a theory of analogical cognition, the conclusion still holds that any model relying on the kind of computational approach to representation that the models I am about to review rely on will necessarily fall short of a deep account of representational development. This also means that if these models of analogical cognition are to be integrated into a more complete computational picture of the mind, they will have to be reconciled with a fundamentally different approach to representation (for example, this does put constraints on the extent to which these models of analogy can provide a deeper explanation of developmental shifts in representation of the kind Gentner *et al.*, 1995 discuss; I will address this further below).

1.1 - Initializing the handcoding framework

Identification of handcoding begins by observing that a model (or class of models) proposed as an account of a cognitive phenomena leaves out some process, entity, and/or function which is hypothesized to be a required part of the SVM that produces the phenomena in nature. As I mentioned in the Introduction, I come to this project with a theory of analogy that cannot be accounted for in current approaches to the computational modelling of analogical cognition. My hypothesis concerning analogy is this: *analogy can involve the production of novel ways of representing, and change in the fundamental representational apparatus already existing* — and this is not just for novel reorganization of already existing concepts of objects and relations, but the *possibility* of creating the concept or representation of a fundamentally *new* kind of object or relation. (This hypothesis is assumed in the *analogical conceptual change hypothesis* of analogical reminding espoused by Dietrich, in press). I argue that present models lack such an account. Furthermore, these models are able to perform as if they can account for all aspects of analogy phenomena because of a reliance on the researcher's involvement in the model (which, I will show in Chapter 4, turns out to be a *necessary* reliance). This researcher involvement allows the model to bypass or ignore the problems of how to come up with novel ways of representing or how to change the current fundamental ways

of representing.

As described in Section 5 of Chapter 2, the two kinds of handcoding — (i) with respect to *researcher involvement* and (ii) with respect to *interpretation* — are actually two kinds of evidence for an explanatory gap, and the circumstances in which that explanatory gap has been imparted to the model. Because handcoding in a model is a result of an explanatory gap in a background theory, handcoding with respect to interpretation tends to be more fundamental, and it is typically the cause of the circumstances that led to inappropriate researcher involvement. This chapter involves demonstrating how two current families of models of analogical cognition are inappropriately handcoded with respect to being able to account for representation emergence and change in that they necessarily rely on the involvement of the researcher, both in the researcher’s direct setup of the model (type 1 handcoding) and in interpretation of representational components of the model (type 2; this latter type, however, will not be completely demonstrated until Chapter 4, where I show how current handcoding is *necessary* for any new representational content within the current approach).

1.2 - The outline and tasks of Chapter 3

There are five main tasks to accomplish in this chapter. First, I will elaborate on the relation between analogy and representation (Sections 2 & 3); this will serve to clarify why I have chosen analogy as the focal point for discussion regarding issues which ultimately have to do with the foundational nature of representation and its role in development. Second, I will summarize the background theories proposed by the families of models of analogy that I will be investigating (Sections 4 and 5); the need for presenting these background theories follows from the criteria which the handcoding framework requires in order to evaluate the computational models proposed in as objective a manner as possible (recall, background theories determine how we are supposed to interpret models, what they refer to or describe, and their ontological assumptions). Third, I will discuss in detail the capabilities and performance of the base family of models and how they rely on handcoding with respect to how representation can emerge and change (Section 6). Fourth, I will discuss some additional details of subsequent models (primarily from SMT) and consider empirical evidence which strongly suggests that representation emerges and changes during the development of analogy-making ability, and the limits these models face in any attempt to offer an account of such emergence and change (Section 7). And finally, I will conclude with a characterization of the kind of representational scheme which these models share in common (Section 8). In Chapter 4, I will show that this common approach to representation is the source of each model’s inability to account for the emergence of structured representation — that is, handcoding has occurred in these models and cannot possibly be avoided within the current framework because of an inherently limiting theory of the nature of representation and the subsequent computational approach to modelling it.

2 - Analogy & Representation

Before turning to the presentation of the models of analogical cognition that I will consider, and how they are hand-coded, the following orienting question must be addressed: Why focus on analogy? Or, to be more specific, this should be split into two questions: (1) What is the relation between analogy and representation? and, (2) If I am ultimately interested in developing a new framework for representation in computational cognitive modelling, why not focus on the computational models of other cognitive capacities, like perception or categorization? Answering either of these requires first addressing some metatheoretical issues concerning the nature of analogy.

A precise definition of analogy cannot be given without a theory. For this reason, attempts at impartiality never start with precise definitions, but instead present a handful of examples of agreed “analogy instances,” and then proceed to taxonomize them according to theory-based criteria. Vosniadou & Ortony (1989, p.2) explain the reason for this practice eloquently:

Because similarity judgements and analogical reasoning are based on people’s representation of entities, even the most superficial treatment of these topics requires some assumptions about how knowledge is represented and how these representations change.

Thus, characterizing a cognitive phenomenon as “similarity or analogy based” bears ontological commitments to representation. Gentner & Markman (1997, p.46) are likewise explicit about this:

... to capture the process of analogy, we must make assumptions not only about the processes of comparison, but about the nature of typical conceptual cognitive representations and how representations and processes interact.

This necessary commitment to particular ontologies of representation serves to answer question (1): Analogy is a *representation-based* phenomena. That is, since how we construe analogical cognition is based on processes involving representation, explanations of analogy depend on how representation is construed, and therefore, the fundamental nature of what representation is will affect our ultimate understanding of analogical cognition.⁴⁴

Of course, the same may be said to be true of a number of other cognitive phenomena, such as perception and categorization. Therefore, my second set of reasons for choosing to focus on analogy (thus, answering question (2)), rather than being a truism about analogy phenomenon, is instead based on the following four arguments:

⁴⁴ In terms of explaining human cognition, I believe the converse of this statement also holds: that the fundamental accounts of analogical cognition will affect the foundational mechanisms at work in representation construction and development — also a key part of my (and others’) decision to focus on analogy.

(a) Analogy is relatively well understood. As Forbus *et al.* (in press) heralds, analogy research represents a “cognitive science success story.” Analogy has a rich history in cognitive psychology and artificial intelligence. The past three decades have produced a multitude of models and associated theories, and we can look back on this history and see a general progression of accomplishments suggesting that real advancement has been made; this is evident in the rich discourse of model descriptions, critiques, and extensions. Furthermore, progress in these fields has come about through a mutual co-evolution of computational models, experimental technique, and background psychological theory (Forbus *et al.*, in press; Hall, 1988; Holyoak & Thagard, 1995). This means that these computational models fulfill the computational cognitive science ideal of not just re-describing a theoretical model in computational terms, but actually playing a role in motivating new theoretical claims, making the model itself an empirical tool to propose and test hypotheses, as well as match performance with the world. (This is not to say that computational research on other cognitive phenomena hasn’t also had similar success; rather, research in analogy also does this, and does it quite well.) Thus, investigating the leading models in research on analogical cognition promises to give us the state-of-the-art in computational modelling and current theories of the nature of analogy. Also related to the point about analogy’s rich history and discourse is the fact that issues relating specifically to representation and handcoding have been raised by Hofstadter (1995) and his research group (Chalmers *et al.*, 1992; French, 1997), and they have made explicit attempts to avoid such handcoding. In the course of presenting the theories and models I will investigate, I will summarize the present status of the debate between Hofstadter’s approach and SMT.

(b) Research on analogy (and similarity in general) is telling us about features of representation that occur over a wide range of developmental levels, from relatively low-level representational complexity in young children, to extremely high-level, adult complexity. Not only do analogy mechanisms seem to be in operation from very early stages of representational development, it is also widely held that roughly the same mechanisms are present from beginning to end of the developmental spectrum, and that there is only change in the complexity of existing representations rather than fundamental changes in basic similarity comparison mechanisms (Gentner *et al.*, 1995; Karmiloff-Smith, 1992; and Goswami, 1992). Analogy research is therefore giving us good insight into how the conceptual ontologies of complex representing organisms have to come to be “structured” in order to accomplish the kinds of reasoning feats that have been observed. Research has led to the widely held belief that analogical cognition clearly involves cognition on concepts which constitute how the agent represents its world (whether representing causal, normative, or imaginative situations about the world). In fact, Gentner *et al.* (1995) proposes that the base mechanism of similarity comparison is instrumental *in* the development of the representation of relational structure that makes analogy comparisons possible (this is referred to as the “relational shift”; I will discuss this in more detail in Section 7.1 of this Chapter, below); Gentner *et al.*, “present evidence that the comparison process itself invites attention to relational structure and ... can promote the acquisition of portable relational knowledge” (p.264). This makes analogical cognition an excellent (perhaps the best) forum in which to address issues of the development of structured representational content.

While I do think that a fundamental re-orientation of the theory of representation

3 - Similarity and analogical cognition

Analogy is taxonomized as a kind of cognition based on similarity (Gentner & Markman, 1995, 1997; Gentner *et al.*, 1995; Hoffman, 1995; Hofstadter, 1981; Vosniadou & Ortony, 1989). As Hoffman (1995, p.18) notes, “Research on similarity judgement has converged on the notion that similarity is rarely just a property of things but is a manifestation of an underlying, context dependent comparison process (Medin, Goldstone, and Gentner, 1993).” And this view of similarity, as Gentner & Markman (1997) note, has important implications for the way we model human thinking, because similarity is demonstrably important across many areas of cognition.⁴⁵

But, what is similarity, what are the different *kinds* of similarity comparisons, and “where” does analogy fall in among them? Tversky (1977) characterizes similarity, in his *contrast model*, according to the number of *features* that two situations or objects share — a similarity comparison is thus a comparison based on the “degree of overlap” of features. *Features*, here, are any kind of attribute that can be predicated of an object or situation; that is, the object or situation can be described as having a certain attribute, like a frog can be described as being green or bumpy (so the features of “green” and “bumpy” are predicated of the frog). Gentner, in her seminal paper, “Structure-Mapping: A theoretical framework for analogy” (1983), notes that while this “degree of feature overlap” model works well for literal similarity comparisons, it does not provide a good account of analogical comparisons. Using her example of the analogy, “an electric battery is like a reservoir,” the illustrative point is made that it neither helps the analogy to point out that batteries and reservoirs both tend to be cylindrical in shape, nor is it weakened by pointing out that they hold very different contents. In fact, a number of features of reservoirs and batteries (like size, shape, color, and substance) are irrelevant to the comparison made in the analogy.

Nonetheless, batteries and reservoirs *do* have similarities. For instance, batteries and reservoirs both store potential energy, and can release that energy to provide power for systems. Thus, what makes these similar is not the number of features held in common, but rather, the kinds of *relations* they hold in common. A relation is somewhat more abstract than a feature in that it describes a kind of association between two or more features or entities in each situation. The storage of potential energy is a relational notion concerning how energy is kept in equilibrium so that it is not turned into power — i.e., about a relationship between a current state of energy, and conditions under which that energy could be turned into power. And the analogy asserts that the same kind of relation holds for both batteries and reservoirs, even though the individual attributes of each are quite different. A simpler example of a relation is found with Gentner’s arithmetic analogy: 3 is to 6 as 2 is to 4. The common relation shared between the 3 and 6 pair and the 2 and 4 pair is that in both cases, the second number is “twice as great as” the first;

⁴⁵ Gentner & Markman (1997), for example, cite a number of different kinds of cognition involving similarity comparisons: we store experiences in categories largely on the basis of their similarity to a category representation or to stored exemplars (Smith & Medin, 1981); in transfer, new problems are solved using procedures taken from prior similar problems; inferences about people are influenced by their similarity to other known individuals (Anderson & Cole, 1990; Read, 1984); and the way we respond affectively to a situation may be based in part on our responses to previous similar situations (Kahneman & Miller, 1986).

The table is to be read by taking the “Few” or “Many” measures of the *attributes* and *relations* columns as designating, roughly, what relative amount of attributes and relations could be predicated of *both* items being compared. The *example* column presents some simple examples of these kinds of comparisons. For example, in literal similarity, “milk is like water” is an example of how both items in the comparison (milk and water) can both have many of the same kinds of attributes and relations predicated of them (e.g., some sample sharable attributes: both milk and water are wet, cold or hot, and have weight; and some sample sharable relations: both milk and water can be poured into a glass, spilled onto the floor, or drunk by a person). However, in the analogy example of “heat is like water,” the compared items (heat and water) have very few, if any attributes that can be predicated of both (unless, of course, you lived in the 18th century), but both certainly share at least one relation: both can be said to “flow through things” (heat through conductive materials, like silver, and water through pipes).

The similarity space is important for my purposes here for three reasons. First, it provides a way of distinguishing how each of these similarity comparisons differ from one another. While the theories of analogy I will be investigating differ in how they go about taxonomizing kinds of similarity comparisons⁴⁶, they generally agree that in analogical comparisons, relations are privileged over features, and I believe that this provides a clear and straight-forward starting point which shows the relation between analogy and similarity more generally. Another nice feature of this taxonomy is that it captures the intuitive notion that there is a continuum along which these comparisons may fall, depending on the situations compared and what kind of emphasis is put on the components of the comparison.

Second, the similarity space demonstrates a couple of key foundational assumptions concerning the kind of representing capability that must be present for certain kinds of similarity comparisons to be made. First and foremost is that people have to be able to differentiate (and therefore, in some sense represent) features in order to make feature-based comparisons, and be able to represent relationships in order to make relationship-based comparisons. Also, it is important to note that features and relations are typically viewed as predicated of (assigned to) entities of some sort, whether they be represented as concrete objects or more abstract, like events, imaginary objects or ideas. In either case, these entities must somehow be represented initially⁴⁷, before these features or relations are associated to them⁴⁸, and thus features and relations require a certain degree of representational structure (representing capacity somehow internally related) — again, a reason why analogy is an interesting phenomena to focus on is because of its prerequisite requirement of structured representation. Furthermore, which kinds of

⁴⁶ French (1995, p.157), for example, argues that this view is too “black-and-white” and that there is much more flexibility in distinctly “analogical” comparisons. With respect to Gentner’s “similarity space,” French’s (1995, pp.13-21) distinctions seem to fall in with what Gentner labels as metaphor.

⁴⁷ Note: this does *not* entail that “whole” entities are represented in order for predication. It may be that only “tracking” visual objects without having to have any sense of object permanence is sufficient for the agent to then “predicate” some feature of it. I will talk about this in Chapter 4.

⁴⁸ Even if these “associations” are merely a tagging of the “location” from the agent’s perspective.

comparisons it turns out people favor should give us insight into how representations are naturally organized internally.

And, finally, the taxonomy exposes a tacit assumption regarding the kinds of processes involved in similarity comparisons — an assumption which a number of researchers have made explicit: that the basic process of carrying out a comparison is the same in each of the kinds of comparison-types taxonomized, and the differences between them rests on the kinds of representations being compared (French, 1995; Gentner & Markman, 1995, 1997; Hofstadter, 1995; Markman & Gentner, 1993; and Medin, Goldstone, & Gentner, 1993).

Like French (1995, p.151-152), I believe it is important to emphasize the *many* different kinds of situations which can involve analogy-making. Namely, there are many “prosaic” analogy-making situations which are still legitimate analogy instances. As French notes, most modelling of analogy focusses on situations which are typically easy to recognize as cases involving analogies. Some examples are proportional or geometric analogies, like those found on IQ tests; scientific analogies, such as the analogy between a model of the atom and the solar system; or sophisticated literary analogies, like those between Cinderella and Juliet, and Macbeth and Hamlet. A number of models have been proposed to account for these interesting cases of analogy-making, and all clearly involve robust representing ability. French, however, appropriately notes that there are many more “everyday” situations in which we make analogies (i.e., we rely on similarity comparisons involving relations), most of which we don’t even acknowledge as such. For example (borrowed from French 1995), a friend sitting across the table from us touches her cheek and says, “You’ve got a bit of spaghetti sauce right here,” and we then go to clean off our own cheek. Or, when shopping in a new grocery store, our companion remarks, “They usually keep the sugar around here.” These are common, everyday occurrences, but are also cases that still require important relational comparisons that must involve representing abilities to some extent — for example, if we did *not* understand that our table-partner’s reference to a part on her body was actually a reference to an analogous position on our body, how could we have ever figured out that the spaghetti was on us?⁴⁹ These are still cases of robust analogy-making and therefore also require accounts of the kinds of representation involved which led to the analogy being made.

(To clarify, this focus on more “lowly” analogy comparisons does not make just *any* comparison an analogy. I follow Gentner *et al.*’s lead in assuming that the key criteria is that analogies are similarity-based comparisons which focus or depend on *relations* shared between situations, in order for decisions prompting certain behavior to be carried out (whether consciously or not). This does entail that there is a continuum of how deeply analogical a similarity comparison is, following from the similarity-space continuum. But as long as relational comparisons are weighted over attribute similarity, as even the above prosaic examples require, I consider the comparison to be an analogical comparison. Analogical comparison processes are importantly implicated in a number of

⁴⁹ From my perspective (my concern of accounting for non-handcoded representation that is adequate for analogy-making), these make up the most interesting class of analogy phenomena precisely because of their prosaic nature, while also requiring a representing of relations — representing that must account for some level of relational structure of a situation.

other cognitive processes — particularly learning and other processes involving memory; and, analogy is also an important component of processes causing knowledge change (Gentner & Wolff, in press), as I will discuss in Sections 6 & 7.)

Among other issues, the claim that there is a central mechanism for analogy-making (whether involving complex representation or more “prosaic” representing and comparison) has also served as a backdrop for asking questions regarding the development of representation. Important research has demonstrated that in a child’s developmental history, the kinds of similarity comparisons they make do change over time; namely, there is a *relational shift* from early reliance on either holistic or object-level similarities to the possibility of purely relational similarities (Gentner, 1988; Gentner & Rattermann, 1991; Gentner, *et al.*, 1995; Goswami, 1992; Halford, 1993). Interesting questions remain open, such as whether the processes themselves change over time, or whether they remain the same and it is instead the kinds of representation available to these processes to work on that need to develop before deeper relational comparisons are made. If the assumption is held that the process is the same throughout development, then the order in which kinds of comparisons are made presents us with a clear picture of what kinds of representations are developing at different times over the course of development (even if the processes themselves change, it is still quite interesting to note the emphasis that eventually comes to be placed on relational similarity). I do believe, following Gentner *et al.* (1995), that it is not only the case that the capacity to make analogies depends on the development of the ability to represent relations, but that the underlying mechanism of similarity comparisons, along with the acquisition of the ability to represent relations, serve to mutually bootstrap one-another to drive representation change and therefore representation development (Gentner *et al.*, 1995, p.264). Again, a story about representation development and a story about the processes underlying analogical comparison seem to go hand-in-hand — I will return to this point again in Section 7.

In the following sections, I will investigate two families of computational models of analogical cognition: the family of computational models based on SMT, and the computational models based on Hofstadter’s (1995) theory of Fluid Concepts and analogy as High-Level Perception (HLP). I have selected these two camps and their computational models based on the following criteria:

- (1) Both families of models are well-defined and have robust expressions of their background theories, and are therefore interesting.
- (2) SMT and HLP have fundamental disagreements on how to study and account for analogy. This disagreement stems from two sources: (a) from a partial recognition by HLP of a kind of handcoding⁵⁰ in the representations of SMT, and HLP’s attempts to avoid it; and, (b) on the other hand, a rebuttal argument by SMT that HLP’s attempts to avoid this kind of handcoding have misled their view of analogy.

⁵⁰ I will distinguish their version of handcoding from how I am using it here.

- (3) And, as I will be concluding in this chapter, *both* sets of models sit squarely within an approach to representation which ultimately fails to avoid handcoding of representation with respect to emergence and fundamental representation change.

While both SMT and HLP have theoretical differences with each other, it is important to highlight that both share a common assumption — an assumption that is shared by practically all computational cognitive science, but rarely interpreted the same way: that the theoretical entities proposed to explain our psychological concepts have a *structure* to them — and the theoretical language used to describe such concepts and their structure starts with the notion of internal (“inside the head”) representation. SMT and HLP disagree on what exactly this structure is, how it comes to exist, and the processes which operate on it to create analogical comparisons; but they both agree that there is *some* kind of structure, and that analogy processes will be explained according to how the processes of comparison, manipulation, recombination and change of representation are sensitive to the “structural” properties of these representations.

It is here that the explanatory metaphor of “representational structure” begins to form. The focus of my investigation will be on how each theory interprets what this structure is and how it is modelled computationally, as such accounts will have profound implications for whether an account *can* be made of novel representational content emergence or change. On the flip side, I also have to take seriously the explanatory metaphor of structured concepts and their explanation by use of the notion of representation. While I disagree with several of the central ontological commitments current approaches to modelling representation take, and I hold differing opinions as to how to explain the metaphor of “structure,” I am still ultimately interested in explaining the same phenomena that these models have observed and modelled. Sections 7 and 8 will discuss what I believe these models do legitimately explain, even if they have deep problems in their approach to representation. This also includes characterizing what it is they have shown that still must be accommodated in a new modelling approach. One of the global goals my development of the situated representation framework is to understand how the kinds of structures which seem necessary for an account of analogy can be accommodated in the alternate framework (e.g., the ability to represent entities, features and relations, and related groups of these).

4 - Structure-Mapping Theory

Following from the above characterization, SMT posits that an analogical comparison determines how the representations of two situations or domains share relational structure despite arbitrary degrees of differences in the objects that make up the domains (Gentner, 1983). But the use of this notion of “relational structure” says more than what I have presented thus far — the emphasis on *structure* highlights the point that simply focussing on the number of shared relations is not enough. As Gentner & Markman (1997) point out, there are an indefinite number of possible relations that an analogy could pick out (Goodman, 1972), but many of these relations are also ignored. Using the example Gentner & Markman give, fishnets and spider webs can be considered analogous for a

number of reasons; for example, both trap prey and both remain motionless while the prey enters. However, other relations, such as the fact that both are smaller than the Taj Mahal, or both are smaller than the Kremlin, would almost never be considered. To answer the question of how we select which common relations to pay attention to, Gentner (1983) proposes the central claim of SMT: that processes of comparison (including similarity judgements — Gentner & Markman, 1997) operate so as to favor *interconnected systems of relations and their arguments*. So, analogy comparisons not only focus on representations of relations, but on connected systems of such relations. The process involved in such comparisons is therefore one of *structural alignment* and *mapping* between mental representations: the relational structure of representation of one domain is systematically compared (aligned) to the existing structures of another represented domain; the greatest “amount” of aligned structure is then proposed as a mapping between representations, thus determining how the one domain is analogous to the other.

4.1 - SMT’s Representational Assumptions

In order to capture the process of structural alignment in analogical comparisons, SMT makes several assumptions about the nature of typical conceptual cognitive representations, and how those representations and processes interact. The primary guiding principle here is a pragmatic one: the representational system must be sufficiently explicit about relational structure to express the causal dependencies that match across the domains; the representational scheme must not only be able to express objects, but also the relationships and bindings that hold between them — including higher-order relations such as causal relationships (Gentner & Markman, 1997).

The first claim that SMT makes about the structure of psychological concepts is that all domains and situations are to be psychologically viewed (that is, according to the mental representation of the perspective of the agent) as systems of objects, object-attributes, and relations between objects. *Objects* are either single entities (e.g., “dog”), components of a larger object (e.g., “dog’s tail”), or even combinations of smaller units (e.g., “pack of dogs”). Despite which of these kinds the object happens to be, they always function as wholes at a given level of organization. Based on this notion of the representational nature of psychological concepts, Gentner assumes that knowledge is then represented as propositional networks of nodes and predicates (a kind of predicate logic format). In this kind of propositional network, the nodes are taken to represent concepts treated as wholes, and the predicates applied to the nodes express propositions about concepts. The predicates can be relations, attributes, functions, logical connectives, or modal operators.

In adopting this scheme of psychological representation of the world as a propositional network consisting of concept nodes related by predicates, Gentner makes two syntactic distinctions concerning predicate types. The first distinction is between object attributes, relations, and functions. Attributes and relations (but not functions) are “truth-functional” in that they are either true of an object or objects, or not true — if the attribute or relation is true, then it holds for the object(s); and if it is false, then it does not hold for the object(s). *Attributes* are taken to be predicates with one argument, and *relations* are predicates taking two or more arguments. For example, GREEN (x) is an

attribute stating that “ x is green,” and ABOVE (x, y) describes a relations of x to y , such as “ x is above y .” Put another way, attributes describe properties of objects (such as being green) and predicates describe events, comparisons, or states applying to two or more objects or predicates. Logical connectives and modal operators, however, are also classified as relations, even though these may also have just one argument (e.g., the logical operator NOT only takes one argument, but is still a relation: NOT (ABOVE (x, y))); Forbus *et al.*, 1995). *Functions*, on the other hand, describe a mapping of one or more objects into another object or constant. For example, HEIGHT (x) does not have a truth value, but instead maps the object x into a quantity describing the height of x . Functions are useful as representational devices because they produce object descriptors (such as HEIGHT (*Clay*) = 6 feet & 3 inches) and may remain unevaluated as the argument of other predicates (such as GREATER-THAN [HEIGHT (*Clay*), HEIGHT (*Thurston*)]) (Gentner, 1983, 1989).

The second syntactic distinction is between first-order predicates (which take objects as arguments) and second- or higher-order predicates (which take propositions or predicates as arguments). An example of this would be a case where two relations are asserted, like COLLIDE (x, y) and STRIKE (y, z). These are first-order predicates which may represent the relations “ x collides with y ” and “ y strikes z .” A second-order predicate may then take these predicates as arguments, thus expressing a higher-order relation: CAUSE [COLLIDE (x, y), STRIKE (y, z)], which asserts that “ x colliding with y causes y to strike z .”

Given these distinctions, Gentner clarifies that these representations are intended to reflect the way people construe a situation, *not* what are the logically possible ways of construing a situation or even the best ways of construing the situation. Logically, a relation may be represented in a number of ways: R(a, b, c) may be represented as Q(x), where Q(x) is true only in the case that R(a, b, c) is true. However, in order to model the psychological make-up of the subject, representations of knowledge structures must be chosen to model the *way* people think about a domain. For instance, R(a, b, c) may be a more appropriate than Q(x), even if Q(x) is defined as being true for all cases that R(a, b, c) is true, because the information that a , b , and c are related in a way R may be psychologically relevant, and Q(x) does not convey this information. Put another way, if the person makes a distinction between a , b , and c and the relation R between them, and does not lump them together under the relation Q(x), then R(a, b, c) is the representation chosen to model that person’s conceptual representation of the situation.

Gentner also clarifies that her theory assumes a kind of relativity — that only relations that apply *within* the domain of discourse are psychologically stored and processed as true relations (as opposed to attributes). For instance, LARGE (x), where x is a cat, asserts a different relative size than if x were a mountain: large cats are always smaller than even very small mountains. Here, LARGE (x) may be interpreted as “ X is larger compared to the average of its class,” and so may be expressed as a relation, such as LARGER-THAN (x , [average member of x ’s class]). The relation between an object and the average member of it’s class, however, is to be interpreted as an attribute of the object, rather than a relation between two different kinds of objects. On the other hand, with two objects in the same domain, a LARGER-THAN relation, like LARGER-THAN (sun, planet), expresses a true relation between two different objects, as opposed to LARGER-THAN (sun, prototype star), which is to be taken as an attribute of a single object. In this

the n^{th} -order relations take at least one $(n-1)^{\text{th}}$ order relation as an argument. Thus, systems can be represented by an interconnected predicate structure in which higher-order predicates enforce connections among lower-order predicates. So a relation or group of relations is more analogically relevant if it represents a larger structure to be mapped to the target. Gentner refers to the principle behind this rule as the *systematicity principle*, which states that a predicate that belongs to a mappable system of mutually interconnecting relationships is more likely to be imported into the target than is an isolated predicate. “People prefer to map connected *systems of relations* governed by higher-order relations with inferential import, rather than isolated predicates” (Gentner, 1989, p.201).

An important feature to note concerning the rules is that they depend entirely on the “syntactic” properties of the knowledge representation, and not on the specific content (e.g., that the node bearing the label “sun” is about a sun) of the domains. This allows for the structure-mapping mechanism which follows these rules to be general. Rather than requiring a specific algorithm for each potential analogy, or even a collection of algorithms for each domain of comparison, the generalized structure of knowledge representation makes it possible for any properly constructed knowledge structure to be compared and considered for structure-mapping. It is this aspect of structure-mapping that is both a boon and a potential liability. The positive side is that structure-mapping is able to consider analogies made in any domain (as long as the “knowledge” of the domain is represented in the appropriate predicate logic structure), which has allowed for structure-mapping to be tested under a variety of conditions, which in turn has led to subsequent refining of the mapping principles based on data collected from the performance of groups of human analogizers. But this explanatory power may have been bought at a high price: acceptance of the structure-mapping account hinges on the plausibility of knowledge being represented according to the scheme outlined above. Current proposed mechanisms which produce these knowledge *structures* that are so vital for structure-mapping can only do so in very limited domains, the building process being entirely dependent on antecedent analysis of what kinds of objects, features and relations there can be. Thus far, no account is given of how new or existing kinds of structures might be learned or built in the first place. This point is further highlighted by the modularity of the structure-mapping algorithm: Gentner’s *architecture for analogy* only assumes knowledge structures may be naturally produced in a more complete cognitive system. I will address this in more detail in Section 4.2 when I consider how current models of SMT are handcoded.

These mapping rules capture three general psychological constraints on the alignment process. The first is that alignment must be *structurally consistent*. That is, alignment must preserve parallel connectivity and one-to-one correspondence between two structures. *Parallel connectivity* requires that matching relations must have matching arguments, and *one-to-one correspondence* limits any element in one representation to at most one matching element in the other representation (Falkenhainer, Forbus & Gentner, 1986, 1989; Gentner, 1983, 1989; Gentner & Clement, 1988; Gentner & Markman, 1997; and Holyoak & Thagard, 1989). So, for example, in the analogy that the atom is like the solar system, an electron corresponds to a planet because they play similar roles in a common relational structure. This, of course, also highlights the already mentioned *relational focus* — that analogies must involve common relations but need not involve

common object descriptions (e.g., it does not detract from the analogy that the electron does not have many of the features that the planet does, like an possible atmosphere or soil composition). The final psychological constraint is made explicit in rule 3: *systematicity* — that analogies tend to match connected systems of relations (Gentner, 1983, 1989). This last constraint is the hallmark of SMT.

Gentner & Markman (1997) offer *cross-mapping* as a striking example of the structural dominance in analogy that is captured by the systematicity principle. A *cross-mapping* is a comparison in which two analogous scenarios contain similar or identical objects that play different relational roles in the two scenarios (Gentner & Toupin, 1986). A simple example of a cross-mapping can be found in the proportional analogy: 1:3 is like 3:9. As Gentner & Markman note, the obvious possibility of matching the two identical 3's is dismissed because to do so would miss-align the relational roles of the terms. Instead, the object correspondences are 1_3 and 3_9, preserving the relational commonality (i.e., the identical ratio: the second number is 3 times as much as the first) across the pair.

(It is important to note that in Gentner's example, the "suppression" requires that the agent employ a kind of *criterial identity* — that the identity given to an entity is based on certain criteria which designate what *kind* of thing the entity is going to be taken as being an instance of. Thus, in the above case, the agent is able to take the '3' not as defined by its being the same cardinality kind in either of the compared situations, but rather, that it plays a role of cardinality *as compared to* its neighbor in the individual situations. As I will present in a moment, the explanation of the mapping already requires that such criterial identity has been decided upon in the way the representation is presented to the mapping mechanism. No account exists yet of how criterial identity is to be applied in the first place. Since such a capacity is so important to cross-domain analogy-making, it seems important to explain how the representations might come to be arranged in the way that amounts to the application of criterial identity. I will return to this point, below, in Section 5.2 of this Chapter.)

Finally, the account of structural alignment and mapping given above allows for another distinction of analogy comparison phenomena: analogical understanding versus analogical reminding (there are other kinds distinctions that can be introduced as well, such as the role of goals, but these two will serve as exemplars to demonstrate the versatility of the structure-mapping framework for accounting for different situations involving the basic analogy comparison process). *Analogy understanding* occurs in a situation in which the cognitive agent is presented with an "already made" analogy; that is, presented with both the target and source (e.g., if I were to tell you, "heat flow is like water flow"). Here, the cognitive agent must then work to understand the analogy — the agent must complete the steps to produce the mapping between the two domains indicated by the presented analogy. These situations can vary in amount of presented information, from minimal information, in which the general domains are simply stated as being analogous and it is up to the agent to figure out what the relational structure of each is that is relevant to the mapping (this usually requires heavy use of memory), to full presentation of an analogy, in which all the intended component relations and elements are given (here, the agent only has to identify the relevant relational structures to make the mapping).

Analogical reminding, however, occurs when one comes up with one's own analogy. Here, it is commonly held that the target is given; for example, the creator of the analogy may be currently thinking about the target domain, or the creator may be presented with a perceptual experience which acts as the target (recall the garbage-can/Stonehenge example from the Introduction). This target representation is held in working memory. The source of the analogy, however, must be somehow retrieved from the analogy-creator's long-term memory. It is widely believed that analogical reminding involves a search of some kind (although, current work on human memory suggests that this is far from the kind of search that our computer does with a hard disk drive, where information is statically and discretely arranged). The problem of finding the source, or seeing that any two knowledge domains are analogous without being cued, is referred to as the *access problem* (Gentner, 1989; Gentner & Landers, 1985; Gick & Holyoak, 1983; Holyoak & Thagard, 1995).

I have raised the above distinction between these two kinds of analogy situations in order to introduce a third analogy-based phenomena — one that might occur in *either* of the above situations: a situation in which the structural alignment and mapping process specifically prompts the *emergence* of a novel way of representing (e.g., the discovery of a new *kind* of relation or entity), or a *change* in one of the fundamental representational predicates or entities on the basis the comparison. This situation, according to SMT (or, at least their computational accounts), *cannot occur*. However, I believe it does — and it is the inability to account for such a situation that I believe is the weakest link in the SMT story. For my purposes here, I am assuming that this does occur, and SMT's inability to account for this kind of situation stems from handcoding, which I will make explicit in Section 6. In Section 7 I will review some Gentner's own empirical data that also suggests emergence and change occurs during analogical comparison. This concludes the basic tour of SMT. I have particularly focussed on SMT's computational and representational presuppositions, and only very cursorily touched on the large corpus of psychological data that this theory has been employed to explain.

4.3 - The family of Structure-Mapping Engine models

The core process of structural alignment and mapping described by SMT has been computationally described in a family of models, all based on a central algorithm called the *Structure-Mapping Engine* (hereafter, SME) (Falkenhainer, Forbus & Gentner, 1986, 1989; Gentner, Falkenhainer & Skorstad, 1987). Several of the other SME-based models which I will briefly describe are MAC/FAC (Forbus, Gentner & Law, 1995), I-SME (Forbus, Ferguson & Gentner, 1994), and Phineas (Falkenhainer, 1987, 1988, 1990a, b).⁵² SME and its relatives have served to test the hypotheses which extend from SMT's central claims and also make more precise the processes for structural alignment and mapping. Again, the theme sounded in Chapter 2 comes to the foreground: that one of the advantages of computational modelling in cognitive science is that it forces the theorizer to make explicit ontological assumptions that may have been previously only

⁵² Gentner & Markman (1997) also note that similar versions of the SME algorithm have been incorporated into other computational models of analogy: Burstein, 1988; Goldstone, 1994; Goldstone & Medin, 1994; Holyoak & Thagard, 1989; and Keane, Ledgeway & Duff, 1994.

implicit. This is likewise true for SMT, as SME makes a precise account of the computational steps for the processes proscribed by SMT as well as the representational scheme necessary for the computations to be carried out. This has also lent to SMT's general explanatory force because of its clarity and the ability to augment psychological theory with working computational examples.

A general important feature of SME's operation is that the mapping-rules instantiated in SME require perfect relational identity for a match to take place (that is, the predicates must be exactly the same). This aspect of structure-mapping is done intentionally and Gentner defends it as necessary to account for two issues important to the mapping mechanism (Gentner, 1983). First, it captures the fact that potential analogies are often missed. And second, it avoids the potential problems of requiring "homunculus-like insight" which would arise if it was required that the algorithm had to antecedently decide how much the comparison mechanism should know about two relations before comparing them. This highlights that how representations are structured is crucial to the operation of the structure-mapping algorithm. This captures the intuition that we seem to all represent things in slightly different ways. The outcome of structure-mapping depends crucially on the structure of the knowledge-representation being compared. Differences in the way things are construed can cause two situations that are informationally equivalent (logically equivalent representations) to fail to match.

This again raises the point that in "taking representation seriously," SMT is not making a commitment to any particular privileged way of construing a given situation in a particular structured form. Instead, it is assumed that the comparison process operates over a person's current representations, however they are derived. Thus, to predict the outcome of a comparison, SME would have to be given the person's current psychological construal of the domains being compared, including goals and contextual information, as well as long-term memory (Gentner & Markman, 1997, p.47). However, while they aren't making any adherence to a particular structure, they have made specific assumptions about the kinds of structures there can be and the fundamental nature of representation — and, ironically, I will be forced, later in this Chapter and in Chapter 4, to conclude that they haven't taken representation seriously enough.

4.3.1 - SME, the core model of the family

The Structure-Mapping Engine is a computer model which simulates the process of structural alignment and mapping, as proscribed by SMT. SME uses a local-to-global alignment process to arrive at a structural alignment of two representations. Operation of SME begins with the mapping-engine being given representations of the source and target domains. The representations may contain varying amounts of complexity or detail depending on how much the "analogizer" that SME is modeling is said to know about either domain. In general, since this is a mapping *from* the source domain *to* the target domain, the source domain will have the greatest amount of detail.

The situation in which the target domain has very little detail in representation, so that a mapping involves the "carrying over" of structure from the source domain, is called *pure carry-over*. Such carry-over transfers structure to the target domain by organizing it according to the nodes matched and the corresponding relations between them carried over from the source (this is to be distinguished from novel structure building — in carry-

Figure 3.6 - Representation of water and heat given to SME

The first stage of the comparison is that of *local matching*. In this stage, SME starts looking for identical relations in the source and target representations. The identical relational pairs found are used to postulate potential matches. For each entity and predicate in the base, it finds the set of entities or predicates in the target that might plausibly match that item. Potential correspondences are evaluated by two simple rules: 1) if 2 relations have the same name, create a match hypothesis; and 2) for every match hypothesis between relations, check for their corresponding arguments — if both are entities, or if both are functions, then create a match hypothesis between them.

From the water/heat flow example, SME follows rule 1 and creates a match hypothesis between same-named predicates, like the GREATER-THAN relation, that exist in both the base and target domains. Again, matches require identical predicates; here, the higher-order predicate GREATER-THAN is exactly the same for temperature as it is for pressure. At this point, two sets are created: a match hypothesis, matching the GREATER-THAN relation with PRESSURE with the one with TEMPERATURE (the first match), and a match for the GREATER-THAN relation with DIAMETER matched with the one with TEMPERATURE (the second match). All these local matches are given *evidence scores* based on the set of local evidence rules, such as the number of shared predicate names, and so forth.

The second stage, *constructing global matches*, begins after a suitable number of local matches have been evaluated and proposed as potential matches in match hypotheses. In this stage, SME collects systems of matches that use consistent entity pairings. To accomplish this, SME first seeks entity correspondences in relational chains in the source and target. When correspondences are found, SME combines them into the largest possible system of predicates with consistent object mappings. These are called *global matches*, and are SME's possible interpretations of the comparison. Associated

then influences what is to be retrieved from long-term memory: the information that is deemed important to the context and is already available to the subject. This creates a potential analog representation (the representations that are given to the structure-mapping engine in the SME model). The analogy engine then produces candidate inferences that are evaluated according to their structure and to the goals and plans of the subject. The evaluator may also influence work of the retriever by having it seek other information in long-term memory deemed missing in the previous analogy processing.

In this way, Gentner separates the planning context from the actual analogy processor. She has done this to identify processes common to analogy across different pragmatic contexts. As described above, structure-mapping is based solely on the syntactic aspects of its knowledge representation scheme. One of the aspects of this syntactic knowledge representation approach is non-contextual functionality, or context independency. This makes structure-mapping the general process of analogy comparisons, not just a method that is used some of the time, or limited to specific domains. Gentner proposes that people use the same structurally guided processor for all kinds of situations that require analogy comprehension, simply adding or removing pragmatic domain-specific constraints on knowledge representation as appropriate. This, she argues, allows the analogy processing to avoid being confined to any particular domain, while at the same time it captures what seems to be the necessary processing that occurs in all domain contexts. (I will address below, in Section 6, whether models that "work within a domain" offer general explanations as well.)

The trick, of course, is accounting for how knowledge comes to be structured in the right way so that when the representations are presented to the mapping engine, they are appropriate for the mapping processes that then take place. It is this point that has drawn the most criticism of the SME model. As already mentioned, however, SME is the core to a more general architecture, which is continually being updated and added to. In sections 4.3.3 and 4.3.4 I briefly review MAC/FAC and I-SME, focussing on the extensions these suggest for the kinds of processes operating on representation. Several other SME-based models, notably Phineas, add some new wrinkles to the structure-mapping story with respect how SMT proposes to computationally account for how representation comes to exist. For this reason, I have chosen to discuss them in Section 7, after introducing the basic issues of handcoding.

4.3.3 - MAC/FAC

At the end of Section 4.2, I distinguished between two general kinds of situations which would involve analogical comparison: analogy understanding (or comprehension) and analogical reminding. SME by itself has been employed to explain analogical understanding. Analogical reminding, however, involves situations in which there is a representation in working memory (the person is currently perceiving or thinking about a situation), and some similar kind of situation is recalled from long-term memory. Here, there is not just a similarity comparison between structured representations; reminding also involves a process of *access* (or "retrieval") of some memory item *based* on the item in working memory. To explain the data that have been found in analogical reminding situation, Forbus, Gentner & Law propose the MAC/FAC system, which employs SME as part of its comparison functions (Gentner & Forbus, 1991, Forbus, *et al.*, 1995).

MAC/FAC stands for “Many are called but few are chosen.” Since the recall from memory is based on its similarity to the item in working memory, MAC/FAC is a model of *similarity-based retrieval*.

MAC/FAC is intended to explain the kinds of comparisons that have been observed to be made in similarity-based retrievals. Gentner and her colleagues distinguish three large classes of reminders (based on the taxonomy of similarity comparisons in the “similarity space,” described above in Section 3):

(1) Analogical reminders — are cases in which only relational predicates are shared. The classic example of this kind is the historically significant scientific discovery made by Rutherford, which is purported to have been motivated by his being reminded of comets while observing behavioral data of alpha particles “shot” at a thin sheet of metal.

(2) Superficial reminders — correspond to the mere appearance comparisons of the similarity space, in which only attributes or features of objects in the compared situations match, but not relational predicates relating the objects. An example of superficial reminders would be seeing a bicycle and being reminded of eyeglasses.

(3) Literal similarity reminders — also called “mundane” reminders, are cases in which *both* attributes and relations of the situations match. An example of a literal similarity reminding would be seeing a bicycle on the street and thinking of your own bicycle (or, remembering that you should put your bicycle inside so that it doesn’t get stolen⁵³).

What makes similarity-based reminders so interesting is that there is a seeming paradox of conflicting drives for retrieval. On the one hand, the type 1 (analogical) reminders are generally the most interesting and can lead to profound discoveries, like Rutherford’s, and can also lead to change or extension of existing knowledge because analogical matches can afford inferences not previously recognized.⁵⁴” And people have been found to prefer analogical comparisons (Gentner, 1989; Gentner & Markman, 1997; and Forbus *et al.*, 1995). But, despite analogy’s high interest, psychological data reveals that most retrieval is of types 2 and 3 (superficial and mundane). So type 1 is rarer than types 2 and 3 (and type 3 tends to be the most common). Thus, it seems that in retrieval, superficial similarity is more important than structural similarity; *but*, the retrieval can’t be entirely based on attribute similarity because *purely* structural reminders (i.e., analogical comparison in which attributes do *not* match) are sometimes experienced (and

⁵³ Dietrich (in press) notes that literal similarity reminders could be the triggers for chains of reminders, not just reminders of literal similar items themselves, like seeing garbage cans and being reminded of your own garbage cans, and that you need to put the garbage out tonight, etc.

⁵⁴ For example, in the Rutherford case, being reminded of comets when observing the alpha-particle data led Rutherford to consider whether the atom really worked like a solar system, in which comets occasionally fly into the solar system, only to be “slung” around the Sun by its high gravitational field and back, almost in the same direction at which they entered; this is opposed to what would be expected if the atom were like a “plumb pudding,” in which objects should be only deflected in different directions, but not “thrown back.” Further inferences could be made on the basis of the comet/solar-system analogy, such as what might correspond to “gravity” in the atom? Is there a “central” object in the middle of the atom that is much more massive than its “satellites” (electrons)? And, are electrons much smaller as well as much further from the central mass than previously believed?... etc. All of these potential inferences for further investigation were unlocked by the reminding that led to the analogy.

I will discuss the SMT account of knowledge change in more detail in Sections 6 and 7.

notably so). Furthermore, that there are such structural-based reminders indicates that sensitivity to structural commonality between the representation in working memory and the representation retrieved must still be preserved to some extent in the account of similarity-based retrieval. The challenge, therefore, is to devise a model of similarity-based retrieval in which literal and superficial reminders occur often, but occasionally analogical reminders are also produced (this latter point entails that the account must maintain the role of alignment and mapping for transfer and inference in similarity comparisons — that it seems unlikely that alignment and mapping would have *nothing* to do with retrieval, since analogical reminders do occur sometimes, even if not often).

Forbus *et al.* answer this challenge by proposing that similarity-based reminding is a two-staged process. In reminding situations, the representation that is in working memory is to be compared to the representations stored in long-term memory in an attempt to find the most promising match. This structured representation in working memory is called the *probe* because it is the one that is being compared with those in long-term memory, in the search for a match.

In order to accommodate the kind of speed that seems to be required, and given plausible constraints on computational resources, the first stage of retrieval, the MAC (“Many Are Called”) stage, does a computationally cheap and quick search through the representations that are stored in long-term memory. This search, however, should not only be efficient, but still preserve some measure of structural information — if the search only focussed on object or attribute matches, then no pure relational mappings could be possible. Forbus *et al.* solve this problem by using simply computed representations of the probe and stored memory representations called *content vectors*. Content vectors are essentially lists of all the predicates that appear in a structured representation, along with an associated number of how often the predicates appear in the representations; listing predicates at least captures aspects of the representations that play a structural role, even if these roles aren’t themselves represented in the content vector. So, for example, going back to the heat-flow/water-flow representations in Figure 3.6 (Section 4.3.1), the predicate logic representation which maintains structural information, compared to the content vector representation, for *water flow* would be:

Structured Predicate Representation	Content Vector
(CAUSE	(CAUSE . 1)
(GREATER (PRESURE (BEAKER) (VIAL))	(GREATER . 2)
(FLOW (BEAKER) (VIAL) (WATER) (PIPE)))	(FLOW . 1)
(GREATER (DIAMETER (BEAKER) (VIAL))	(OBJECTS . 4)

(Note that specific object kinds or proper names are dropped, being treated only as entities which are predicated as being objects — and thus enumerated only as OBJECTS.) A content vector is assumed to be created for and remain associated with each structured representation item in memory; this is true for the probe as well. A search through memory then entails that the probe’s content vector is compared with the content vector representation of each of the items in long-term memory. A comparison results in a score for each particular probe/memory-item pair; the score is computed by taking the dot product of the content vector of the probe and the memory-item. Once all of these scores

are computed, the MAC *selector*, an algorithm which selects the best matches, takes the best match and everything within 10% of it. The memory-items of each winning match are then sent to the second phase of the retrieval process.

The second phase, the FAC (“Few Are Chosen”) phase, is essentially a bank of SME matchers, all running in parallel in literal similarity mode. Running in literal similarity mode leaves the SME matchers open to the possibility of computing relational similarity, object similarity and overall similarity, thus allowing for reminding situations of different kinds of similarity, just as people are capable of: the possibility of types 1, 2 and 3.⁵⁵ These matchers take as input the memory descriptions that are passed forward by the MAC stage and compute a structural alignment between each of these descriptions and the probe. This comparison works essentially the same as the SME demonstration above in Section 4.3.1. While the MAC stage primarily focussed on a simple and quick method for finding what memory-items might share in common with the probe (while ignoring specific structure), the FAC stage captures the human sensitivity to structural alignment and inferential potential (subject to the limited and possibly surface-heavy set of candidates provided by the MAC stage). Once all of the SME matchers have produced structural mappings, a FAC *selector* chooses a small set of matches (based on maximal structure) for additional processing (such as proposing candidate inferences, etc.).⁵⁶ This completes the analogical reminding process.

The MAC/FAC model has been successful at replicating (predicting) human data (Forbus *et al.*, 1995). It is also an interesting extension of the basic SME algorithm, demonstrating its flexibility and applicability to similarity-comparison situations beyond simply presentation of the two representations to be compared. The main point to made here is that MAC/FAC relies on the same basic SME-style representation. The content vector is an interesting derivative measure of features of structured predicate representations, but does not entail a departure from the basic predicate calculus representation scheme.

4.3.4 - I-SME

Another variation of the basic SME algorithm is capable of *incremental mapping*: I-SME (Forbus *et al.*, 1994). I-SME is an extension to the SME algorithm in the sense that it has all the capabilities of SME, including the ability to generate novel candidate inferences, except that I-SME has the additional capability of extending its existing interpretations when given new information about the source or target. Information can therefore be presented to I-SME all at once (in this case, I-SME’s results are identical to SME’s), one fact at a time, or any other desired combination. The advantage of incremental mapping is that SME can be extended to account for many cognitive tasks

⁵⁵ If the SME matchers were set to favor high-level structures only, then this would result in very few surface similarity results (type 3), which is exactly the opposite of actual data: type 3 are the most common, followed closely by type 2, and then relatively few type 1.

⁵⁶ A number of different “selector thresholds” are possible. Currently, MAC/FAC is favored to have it’s FAC selector choose the top 10%, as this usually only returns one representation; on occasion it chooses more, but in these cases the selected structures are so close that they are all fairly relevant, and this seems plausible for human recall as well.

involving analogy, such as understanding metaphors, problem-solving, and learning, which require the ability to extend mappings as new information is found.⁵⁷ I-SME does not account for how that new information might be sought — only what to do when it is added (although mappings can now be utilized by programs using SME, such as a problem solver which could have as a goal extending or verifying a particular mapping proposed by the current state of I-SME). The primary changes to the SME algorithm are that existing mappings can be extended (rather than starting from scratch) as new information is added to either the source or target, and a *remapping* operation provides backtracking at the level of mapped components if the accumulated mappings become suboptimal. Also, I-SME can be guided by external constraints, such as a *required* filter — this filter serves to force externally imposed correspondences (e.g., if the analogizer is told that “heat is like water,” it will always preserve mapping pertaining to heat being like water). The main interest of I-SME here is to highlight how the SME algorithm has been extended to be flexible even to the flow of information that is presented to it. Also, as Forbus *et al.* (1994) note, the I-SME extension brings this increased flexibility with little increase to computational complexity, maintaining the structure-mapping algorithm’s speed and efficiency. Again, the approach to the nature of representation remains the same.

5 - Conceptual Fluidity and Analogy as High-Level Perception

I highlighted at the end of Section 3 that there is a universally shared assumption concerning theoretical entities used to explain our concepts: that our concepts have some sort of *structure* to them which describes or determines their behavior and role in cognition. The question before us as cognitive scientists is, then, what’s the nature of conceptual structure, and what are the processes that operate on or within these concepts to produce the behavior that we observe. In computational cognitive modelling, these questions are addressed in terms of the treatment of representation. It is here that we find the biggest underlying difference between SMT and Hofstadter’s (Hofstadter & FARG, 1995; Chalmers *et al.*, 1992) theory of analogy as *High-level Perception* (HLP): a differing intuition about how to characterize the behavior of concepts in order to account for distinctly human intelligence.

For Gentner’s SMT, the focus has been on identifying the structure of representations and the processes which operate on those representations, based on their structural properties, to produce cognitive phenomena (such as analogical comparisons). There has, of course, been a mutual evolution of how representational structure is construed and what processes are proposed to work on these structures. Nonetheless, SMT’s underlying distinction between representation and cognitive processes which operate on them remains the same.

Hofstadter’s central focus, however, has been explicitly aimed at capturing the *flexibility* and *creativity* of human intelligence. This difference in starting points has

⁵⁷ I-SME is used in two SME-based systems: MAGI, a simulation of symmetry detection (Ferguson, 1994), and MARS, a model of analogical problem solving, in which I-SME imports equations from previously-worked thermodynamics problems to help in solving new problems (Forbus *et al.*, 1994).

motivated a difference in methodological approach and theoretical focus. In effect, Hofstadter and his *Fluid Analogies Research Group* (FARG) tackle the issues of the relation between representational structure and cognitive processes from the opposite direction: rather than asking, “What is the structure of representation and the processes which operate on it?” he asks, “What are the underlying dynamics of concepts and how they are involved in the construction of the representational structure — such structure *being* the end-result of the cognitive phenomena?” It is important to note that there is still much that SMT and HLP agree on; namely, that analogy does involve an alignment and mapping between structured representations of two situations (i.e., a mapping between structures of related entities). The proposed processes by which this occurs, however, are significantly different. In the HLP approach, *how* representations are constructed — the process involved in the identification of entities and their relations to arrive at structured representation — is the key to the story of how the structured representations of two situations come to be aligned and mapped. For this reason, the HLP approach adds that there is also the process of *situation perception*, in addition to the process of mapping, and argue that these two situations are not separable.

According to HLP, our picture of the dynamics of representation itself needs to be deepened and intricately woven into the process of analogy-making in order to capture the observed dynamics of analogical cognition. For Hofstadter, the question of where to turn in order to capture the dynamics of representation construction that SMT misses is in *perception*: for Hofstadter, perception *is* the process of constructing representation, and perception and high-level cognition are inseparable (or, more specifically, the high-end of perceptual processes *is* high-level cognition): hence, analogy is a kind of high-level perceptual process.

FARG first formally introduced their theory of analogy as high-level perception in the Chalmers *et al.* (1992)⁵⁸ paper, “High-level perception, representation, and analogy: A critique of artificial intelligence methodology” (although the general cognitive architecture which underlies this approach dates back to the early 1980’s; Hofstadter & FARG, 1995, is the most comprehensive summary of the FARG research project). The theory of high-level perception is based, in part, on the insight Kant had regarding our “understanding” of a situation: that a cognitive agent understands what it is perceiving as a result of the organization of raw sensory data into a meaningful form on the basis of the cognitive agent’s concepts about the world. The main thesis of the Chalmers *et al.* paper is that high-level perception is deeply interwoven with other cognitive processes and current AI models that ignore this fact miss the dynamic nature of cognition as involved in, for example, analogy-making.

5.1 - High-level perception and flexibility

Low-level vs. High-level perception

According to Chalmers *et al.*, perceptual processes form a spectrum which may be divided into high-level perception and low-level perception. Low-level perception

⁵⁸ I will refer to this paper and its authors, David Chalmers, Robert French and Douglas Hofstadter, as “Chalmers *et al.*” from here on.

begins, roughly, with the reception of raw sensory information by various sense organs and ends at the point where high-level concepts begin to influence organization of sensory information. Processes that occur early in the visual system are an example of low-level perception: information from the rods and cones of the eye is passed up the optic nerve, where it is processed in the lateral geniculate nuclei, and then passed to the primary visual cortex and superior colliculus. In this area from the lateral geniculate nuclei to the visual cortex and superior colliculus there is processing of brightness contrasts, light boundaries, and edges and corners in the visual field, as well as some location processing. The processing going on in these areas is important for building the foundation upon which representations used in higher-cognition will be based.

High-level perception begins at the level of processing where *concepts* begin to play an important role. High-level perception may be subdivided into a spectrum ranging from concrete to abstract processing. That is, the spectrum ranges from object recognition (e.g., being able to recognize an apple), to the ability to grasp relations (e.g., being able to determine the relationship of a cat to a mat: “the cat is *on* the mat”), to the ability to grasp abstract relations (e.g., being able to grasp that “Bill Clinton is *in* the Democratic Party”). The most abstract kind of perception is processing of entire complex *situations*, such as a love affair or a war. In high-level perception, the processes involved are distinct from particular sensory modalities.

Flexibility

Chalmers *et al.* claim that one of the most important properties of high-level perception is that it is extremely *flexible*. This flexibility is observed in how a variety of data are perceived in a number of different ways, depending on the context and state of the perceiver. According to Chalmers *et al.*, flexibility is a key feature of high-level cognition, and because of this flexibility, “it is a mistake to regard perception as a process that associates a fixed representation with a particular situation” (Chalmers *et al.*, p.187). That is, probably no single representation is adequate for any given situation; thus, the acquisition *and* processing of structured representation must be flexible. They posit both context and top-down cognitive influences as responsible for flexibility in representation and cite four main sources of flexibility (including examples):

(1) Perception may be influenced by belief — our expectations play an important role in determining what we perceive, even at quite a low level. For example, a man arriving home to find his wife on the couch with a stranger might have entirely different perceptions of the situation depending on whether he suspects that his wife is cheating on him or believes that an insurance agent was scheduled to meet her that day.

(2) Perception may be influenced by our goals — objects may be perceived differently according to what the perceiver is trying to accomplish. For example, consider two different possible perceptions of a log on a trail depending on goals of the perceiver: if hiking, the log may be perceived as an obstacle; whereas, while attempting to build a fire, the log may be perceived as firewood.

(3) Perception may be influenced by external context — For example, consider how the figure ‘A’ looks different in the following contexts (Figure 3.9):

- (2) The problem of organization — deciding how these data are put into the correct form for the representation; that is, even if we know which was the framework for construing mental representations, there is still the problem of organizing the data into the representational form in a useful way.

These two problems are taken by Chalmers *et al.* to form the basis for the requirement of cognitive models incorporating high-level perception. This also serves as the starting point for the identification of what Chalmers *et al.* argue has amounted to reliance on handcoding in current models of analogy. They argue that in the traditional approach, the data are organized by the human programmer who appropriately fits them into the chosen representational structure. To do this, researchers usually have to rely on *their* prior knowledge of the nature of the aspect of cognition being modeled in order to *hand-code* a representation of the data in near-optimal form. Only after this hand-coding is completed is the representation allowed to be manipulated by the machine, independent of outside influence. In this way, Chalmers *et al.* claim that the “real” problem involved in determining representation, and therefore the problem of high-level perception, is ignored (Chalmers *et al.*, 1992, p.189). I have already presented an example of this kind of handcoding in Chapter 1: BACON, the model of scientific discovery; my analysis of BACON is based on the analysis made by Chalmers *et al.*

(It is important to note here that while these problems are worded such that high-level perception appears to be the only way of achieving better cognitive models, Chalmers *et al.* have not addressed the question of whether there can be dynamic representation without incorporating perception. In other words, they may have created a false dilemma; or their definition of what counts as high-level perception may be any representation that is dynamic enough for the production of analogy, which begs the question.)

The *real* problem before AI is then, not what is the proper structure of representation, but how such representation could be arrived at, starting from environmental data: the task of understanding how to draw *meaning* out of the world. As Chalmers *et al.* put it, there is a *meaning barrier* which has rarely been crossed by work in AI:

On one side of the barrier, some models in low-level perception have been capable of building primitive representations of the environment, but these are not yet sufficiently complex to be called ‘meaningful’. On the other side of the barrier, much research in high-level cognitive modeling has *started* with representations at the conceptual level, such as propositions in predicate logic or nodes in a semantic network, where any meaning that is present is already built in.

(Hofstadter & FARG, 1995, p.174).

Chalmers *et al.* are quick to recognize that one might defend such handcoding by arguing for the possibility of a representation module. What this amounts to is positing the possibility of leaving the problem of representation-formation until later. What is implied by this attempt at not addressing issues of high-level perception is the tacit assumption, “that it is possible to model high-level cognitive processes independently of perceptual processes.” Chalmers *et al.* are sceptical of the feasibility of a separation of

perception from the rest of cognition. For one thing, a representation module is unlikely to produce the single “correct” representation for any given situation. To attempt this would be to miss the flexibility observed in human perception. For such flexibility to arise, the representational process would have to be sensitive to the needs of all the various cognitive processes in which they might be used. (French, 1997, adds additional detail to the argument against the plausibility of a representation module.)

Chalmers *et al.* are correct in that it seems unlikely that a single representation would suffice for use in all cognitive processes or for all situations for a single cognitive process. Since, as they argue, high-level cognition is so flexible, representation must be able to vary with different contextual and top-down influences. This is directly counter to the “representation module” hypothesis they propose because the representational module hypothesis suggests a representation-building system that is *separate* from such context and top-down influences.

Chalmers *et al.* claim that because of the untenability of a separate representation module on the grounds that it would require the necessary representation structure prior to any processing, an account must be given in which representation building is *entirely* integrated with other cognitive mechanisms. Because of this, they conclude that separating representation-building from higher-level cognitive tasks is impossible. They claim that any fully cognitive model will require a continual interaction between the process of representation-building and the manipulation of those representations for a particular task in order to get the kind of flexibility that is apparent in high-level cognition. If this is true, then continued modeling based on the “representation module” thesis will eventually, as Chalmers *et al.* put it, “... lead up a dead-end street” (Chalmers *et al.*, 1992, p.191).

(Again, I must flag another false dilemma. The underlying assumption here is that the “representation module” would have to somehow work completely independently of any other processing — be completely encapsulated. Why can’t a representation module interact with current processing of representations by other modules? This would then allow representation construction processes to influence and be influenced by other cognitive processes, while still remaining modular. I will return to this point again in a moment.)

As an example of high-level cognition that depends on high-level perception (and *vice-versa*), Chalmers *et al.* take up the cognitive process of analogy. Chalmers *et al.* divide the processes involved in analogical cognition into two basic components. The first is the *situation perception*, in which the data involved in a given situation are filtered and organized according to a given context. The second, *mapping*, involves taking the representations of the two situations and finding appropriate correspondences between components of one representation with components of the other to produce the analogy match-up. Chalmers *et al.* claim that, “it is by no means apparent that these processes are cleanly separable; they seem to interact in a deep way” (Chalmers *et al.*, 1992, p.195). They propose that because perception underlies analogy, we are tempted to divide the process of analogy sequentially into situation perception, followed by mapping (as Gentner’s analogy architecture suggests). However, analogy is deeply involved in the situation perception stage as well; perceptions of many situations are possible because of analogical mappings. Thus, Chalmers *et al.* conclude that situation perception and mapping processes go hand-in-hand. Although the situation perception process is

conceptually prior, as the mapping process requires some representation to work on (thus, it is ‘grounded’ in a preceding perceptual process), perceptual processes may also involve analogy in the further construction of the representation of a situation. Put another way, the interdependence is expressed in that perception produces a particular structure for the representation of a situation, and the mapping process emphasizes certain aspects of this structure, which in turn influences what further structures are built.

For these reasons, Chalmers *et al.* propose that perception must be accounted for in a model of analogical thought: what is needed for a deep understanding of analogy is a model in which the processes of perception and mapping are not theoretically separable — something they claim has not been taken into account by current models of analogy. They offer, based on preliminary arguments above, two arguments for why the situation perception and mapping processes aren’t separable.

The first is that perception is dependent on the processes of analogy: people interpret (perceive) new situations in terms of old ones. When they do this, they are using the analogical mapping process to build up representations of various situations. Chalmers *et al.* give an example of this interpretation process: most Americans condemned the death threats made to the author of the controversial book, *The Satanic Verses*. However, some conservative Christian leaders reacted differently, feeling sympathetic to the Iranian Moslems, comparing the Moslem’s situation to their own when the controversial (and considered blasphemous) film *The Last Temptation of Christ* came out. Their perception of the situation was significantly altered by this salient analogy with their own past. Because of this, Chalmers *et al.* conclude that it is therefore impossible to split analogy-making into “first perception, then mapping” — such an approach that did simply would not be able to account for how two radically different perceptions could arise from one in the same set of data; this context dependence based on an analogical relation seems to necessarily occur in how a situation is perceived.

I should digress for a moment and note that SMT, which Chalmers *et al.* accuse of taking a modular approach to representation (and therefore shouldn’t be able to account for this phenomena), has a natural account of this situation: namely, it is not the case that SME has to have just the right perception of the situation in order to make an analogy. In fact, the SMT argument is that the analogical comparison above was not a case of perception, but a case of *reminding*. And this is the kind of phenomena that MAC/FAC is a model of. Consider two MAC/FAC models, one with the memory of a conservative Christian leader, the other with the memory of a more liberal-minded religious person. The SMT-based argument goes that both could be presented with the same data (which could be fed in as the *same* representation). The key difference between the two conceptions of the situation then arises in the kind of analogical mapping that might be made based on the two people’s background knowledge — their memories. In the first case (the conservative), the representation of *The Last Temptation of Christ* situation maps well with the Iranian Moslems’ situation, while the in the second case (the liberal), no such parallel is drawn because the memory of *The Last Temptation of Christ* simply isn’t comparable to the representation in working memory. Or, it may even be possible that the liberal might be reminded of the conservatives’ reaction to *The Last Temptation of Christ*, but they themselves ultimately don’t identify with similar kinds of conservative beliefs, and so only consider it an analogous situation *for* conservatives. Chalmers *et al.*’s argument doesn’t even consider whether it might be possible for liberals to be

reminded of the situation that the conservatives were, but nonetheless see the analogous situation from a different perspective — a non-sympathetic position. The MAC/FAC explanation could additionally argue that why the conservatives were more apt to identify with the memory of the previous situation (and perhaps more readily recall it) was because when they experienced the blasphemous movie situation, not only were they in a similar kind of situation as the Iranian Moslems, but this situation was also highly emotionally charged, and therefore much more likely to be prevalent in memory. The liberals' experience of the movie situation, however were from the perspective of outside observers and therefore more detached.

The MAC/FAC account can just as well explain the *same* data that Chalmers *et al.*'s first argument depends on. This is not to say that the two models say the same thing — they are clearly different proposals. The key difference boils down to SMT relying on memories, representations stored in long term memory, and reminding is a matter of retrieval, whereas Chalmers *et al.* believe the memory is wrapped-up *in* the act of perception, in making sense of a situation. (It would be interesting to know how fast it was that the Christian conservative saw their situation as analogous; Did it happen *as* they came to see the situation as anything at all, or did they perceive the Moslem's situation, and *then* were reminded of their own analogous experience?) I will return to this difference in views below. The main point here is that this first objection is not very compelling as a knock-down argument against SMT's modular approach and the computational models based on it.

(The deeper question for SMT, I believe, is why the identities of the relations in the compared representations (both the one of the Iranian Moslems' in working memory and the recalled representation of *The Last Temptation of Christ* situation) — identities required in order for any recall or mapping in MAC/FAC to take place — antecedently match in the first place? What's the guarantee for antecedent identity? It is here that Chalmers *et al.* might rebut that the act of perception is what *makes* these situations mappable — is what makes the requisite identities that Gentner then claims are mapped for the analogy. Interestingly, I believe Gentner's answer would be that there *is* a kind of situation perception which takes place to arrive at the current representation in working memory. But, rather than being a complete constructive process, it is a rather quick construction on the basis of current conceptual state. All the same, it is unclear that this alone provides Gentner with the identity-guarantee required (Dietrich, in press) — I will return to this point below.)

The second problem that Chalmers *et al.* raise (the second argument) concerns the modular approach to separating representation construction from analogical mapping processes: analogy depends on representation construction processes. Chalmers *et al.* argue that this view stems from the conception of analogical thought as something separate from the rest of cognition. Here, analogy is a special tool in reasoning or problem-solving, employed only some of the time, but always after the perceptual process has created the representation. If such a representation module existed, it would have to supply a single “correct” representation for any given situation, independent of the context or the task for which it is being used. This representation would have to provide all possible information for each representation.

For example, consider two possible different comparisons between (i) DNA as a zipper and (ii) DNA as source code. The DNA representation would have to include both

structural information as well as information-carrying properties; this would be a single representation incorporating information about its physical, double-helical structure, about the way in which its information is used to build up cells, about its properties of replication and mutation, etc. The size of this representation would make it unwieldy for use in higher-level task-oriented processes for which it was intended. Thus, computer models, such as SME, operate with relatively “small” representations that have tailored information relevant to the analogy. If much larger representations were needed to account for all available information necessary for any situation, there would need to be radical change in the SME program design. The solution to this problem inevitably leads to integration with high-level perception:

The problem is simply that a vast oversupply of information would be available in such a representation. To determine precisely which pieces of that information were relevant would require a complex process of filtering and organizing the available data from the representation. *This process would in fact be tantamount to high-level perception all over again.* This, it would seem, would defeat the purpose of separating the perceptual processes into a specialized module.

(Chalmers *et al.*, 1992, p.200)

Given the seeming unavoidability of there being an interplay between perceptual and mapping processes in order to capture the flexibility of representation that arises from context and particular tasks, Chalmers *et al.* conclude that it is now necessary to investigate how perceptual and mapping processes can be integrated. (Note that Chalmers *et al.* and FARG are somewhat ambiguous about the necessity of this integration for any account; one reading of Chalmers *et al.* is an argument for an *in principle* necessity of interweaving of the representation construction and alignment and mapping processes — in the next sub-section, I will investigate problems with this strong claim.)

Some problems with the Chalmers et al. approach to handcoding, and SMT's rebuttals

I do believe Chalmers *et al.* have raised some important points. In particular, they have noted that having to have the representations with structure be handed to SME prior to any structural alignment and mapping *does* constitute an identification of a kind of handcoding in the use of a computational model to explain analogical cognition: the lack of an account of the role of the dynamics of representation manipulation and construction *in* the process of analogy making in models like SME does entail missing an explanation of such dynamics as they appear in observed analogy-based behavior (although I will discuss in Sections 6 & 7 some important additions to the SMT account).

I think a good example of what is missing in the SME computational models, a phenomena that is even frequently discussed by Gentner (e.g., Gentner, 1983, 1989; Gentner & Toupin, 1986; Gentner & Rattermann, 1991; and Gentner & Markman, 1997), is the cross-domain comparison situation, such as the example I described near the end of Section 4.2: the analogy “1:3 is like 3:9” is mapped with the 1 in the first group aligned with 3 in the second group, and the 3 in the first group aligned with the 9 in the second

group, in order to favor the relation between the two numbers in each group (the relation being: the second number of a given group is three times as much as the first number). Thus, to make the analogy, one must “suppress” the attempt to align the two 3’s which happen to appear in either group. As should now be clear, SME’s success at making this “suppression” results from the fact that SME just ignores the identity of items which are the subjects of relational predicates.

But, as I mentioned at the end of the description of the cross-domain example in Section 4.2, it seems important to understand *how* the representation came to be arranged in this way so that the proper alignment can take place. Again, when attempting to decide what to make of the identity of the components of the presented problem, it seems very important that the *application of criterial identity* to the components is accounted for: that is, *why* we should choose to structure our representation of the situation so that the relation of the 3’s to their neighbors is favored over the simpler identity of the two 3’s to one another? It seems we make a decision to do so (otherwise how could it be that we *can* still see it either way), and it is this decision to structure our representations this way which makes the analogy possible. Therefore, this decision is very important to account for in explaining the observed analogy phenomenon. This is a good example of how handcoding where and how representations come to be structured leaves out some crucial aspects of the analogical comparison process. And this is the kind of handcoding that I believe Chalmers *et al.* are interested in exposing (although they don’t use this example).

Chalmers *et al.*, however, appear to draw the particularly strong conclusion that such handcoding entails a complete failure of SME to explain anything about analogical cognition. This seems wrong, and is one of several additional points where my use of the handcoding identification framework to critique SME makes a distinct break from Chalmers *et al.*’s use of the notion of handcoding.

First, a technical point: Chalmers *et al.* identify SME as among a class of models which are based on the methodological goal of finding *the perfect representation*, which will then supply all that is needed to make an analogy. This is a misunderstanding of SME’s treatment of representations. SME is explicitly slated (Gentner, 1983, 1989; Gentner & Markman, 1997, p.47) to work on *psychological* representation: the representations which are *currently* in the agent, as already described at the end of Section 4.3. This, of course, does not mean that SMT evades the charge that no account is given of *how* these psychological representations come have the form they do — but at the same time, SMT is *not* arguing for “ideal representations,” as some other AI projects have explicitly searched for.

Second, my position is that handcoding in a model simply highlights what it does not explain — and in the process, helps to make clear and define what it is that is missing so that a new model can be proposed which accounts for the previously lacking mechanism (what was previously supplied by the researcher so that such an account in terms of an actual mechanism in the model was not required for the model to still perform). While in some cases such handcoding may mean that the model is entirely incorrect about its account, this is not a necessary entailment following from the identification of handcoding. Instead, in most cases the model that has relied on handcoding may miss some important pieces to the story of how the phenomena is produced, but does still account for other aspects of the phenomena. As already mentioned, even the HLP models propose mechanisms that accomplish structural alignment and mapping — even

if they're mixed in the process of constructing representation out of structured representation pieces.⁵⁹

This leads to a third point: I have stressed that in the identification of handcoding, it is important to consider the background theories and related models proposed by a theory in question in order to properly evaluate its proposed model. Forbus *et al.* (in press) have recently defended their methodology and use of SME in explanation, noting that while SME does not integrate issues regarding the dynamics of the construction of representation from constituent representational parts in the process of making an analogy, this issue has not been ignored by SMT. As already mentioned, the I-SME and MAC/FAC models extend SME's use to account for the additional kinds of flexibility that appear to arise in representation construction and processes involved in the analogy-making process, such as the role of stored representations in long-term memory and the addition of new information while mapping. Gentner, in fact, has also been keenly interested in this representational development issue. Gentner and Laura Kotovsky (Kotovsky & Gentner, 1990) have investigated how children progressively make finer distinctions in their use of descriptive terms, therefore "unpacking" the details of a terms meaning. I will discuss this, and the developing SMT account of knowledge change (Gentner & Wolff, in press) in Sections 6 and 7.

This is not to say that the HLP and SMT accounts are identical — they are both still clearly different proposals for an account of the dynamics of representation structure manipulation and change *in* analogy processing. For example, the depth of HLP's account of representation construction through the involvement of high-level concepts still seems to extend beyond SMT's account of alignment and mapping on the basis of two compared representations. The treatment of the flexibility of long term memory made in the HLP models does seem more plausible than the data-based view SMT proposes, where long-term memory is treated as a repository of representation structures. Memory appears to be much more dynamic than even an I-SME plus MAC/FAC system proposes (see Barsalou, 1983, 1987). Oshima (1996) explicitly addresses this issue of theories of memory related to the models proposed by SMT and HLP; he provides a number of arguments that support the view that concepts in long-term memory are malleable and changeable, in the way HLP argues for conceptual fluidity (below), not rigidly represented in the way MAC/FAC proposes.

Nevertheless, it is not entirely accurate (at least anymore) to accuse SMT of handcoding while *only* considering the basic SME model. With the addition of the other SMT models aimed at explaining some of the dynamics that HLP has claimed are a result of representation construction during analogical processing, Forbus *et al.* defend their modular approach to the study of analogical cognition by arguing that Chalmers *et al.* indeed posed a false dilemma (as alluded to above) in proposing that the process of

⁵⁹ "How much" is central to or required for a full account of the phenomena in question is a conceptual and empirical question ultimately having to do with the identity of the phenomena. We will only have a definitive answer in retrospect, and only if we did have such an answer could we claim that certainly such handcoding entailed missing entirely how the phenomena works. For example, here I am only advancing the claim that representational content emergence and change is implicated in analogical processes, and that current models lack an account of this. I am not claiming that such models therefore don't tell us *anything* about how analogical processes work, nor anything interesting about properties of cognitive representations.

construction and manipulation of representational structure is either serially modular or entirely integrated. In breaking this dichotomy, Forbus *et al.* propose that a third and perfectly viable alternative is to posit the modularity of different analogical processes, but that the modules are continuously interacting with one-another; thus, analogical cognition is interleaved with perception, but the two are still separable processes and can be studied independently. The advantage of this modular approach is that it is possible to consider the individual constraints and mechanisms in each module, thus getting a better picture of the detailed operation of analogy as a whole. This modular decomposition is therefore useful in investigation and explanation. While the possibility of decomposition is not precluded in a “fully-integrated” model such as HLP’s, it does lead to conceptual and empirical clarity (Forbus *et al.* correctly note that FARG routinely explains their models by decomposing them into modular functions, thus picking away at the Chalmers *et al.* claim that the processes are somehow *in principle* not separable).

At the same time, it should also be noted that at some point these modules *do* need to be integrated into a “complete” whole model of cognition — the distant goal of a “complete” cognitive science model of the human cognitive system. While the modular approach seems important, if not necessary to understand the detail of individual functional components of cognitive processing, it is also important to simultaneously work on the project of integration so that the processes which are a product of a number of component processes can be understood; and this is evidenced in SMT’s proposal of the I-SME model, which seems much more robust as a general analogy-mechanism than the original SME model which had no capacity for an account of representation modification *during* analogy processing.

The reason for raising these points, aside from noting that one should not over-extend the impact that recognition of handcoding may have on a critiqued model, is that I also need to be clear about what methodological role handcoding recognitions play. I agree with Forbus *et al.*’s point that decomposition is a necessary and revealing way of studying cognition. And the present state of the SMT vs. HLP debate demonstrates that both approaches have their merits. However, while some processes are decomposable, that does not mean that certain decomposition is possible with *all* phenomena — *and*, some decompositions are better than others. As I will be arguing at the end of this Chapter and Chapter 4, handcoding with respect to representation emergence and change has led to particular limiting approaches to the fundamental nature of representation — and changing such approaches to representation will have a profound affect on how representation is handled in both the SMT and HLP models. While both SMT and HLP models offer interesting insights into processes of analogical cognition and the role representation plays, a deeper account representation dynamics as involved in analogy-making requires not just integration of modules, but recasting processes that involve representation.

5.3 - The architecture for conceptual fluidity and HLP

Slippage and the relational structure of concepts

Given the metaphor of high-level perception for describing the intimate involvement of concepts in the perception of a situation, and the emphasis put on the highly flexible

nature of such representation, FARG has proposed an outline of the primary features which a model of concepts should account for. This, in turn, has laid the foundation for the architecture for conceptual fluidity and high-level perception.

This outline begins with the notion of conceptual “slippage” (introduced in Hofstadter, 1979), which describes the situation in which a concept employed in describing some situation is replaced by a related, but different concept. The paradigm example of conceptual slippage frequently cited by FARG is the range of analogies that could be produced in considering the following question: During Margaret Thatcher’s tenure as the head of state of Great Britain, who was the “first lady” of Britain? (French, 1995; Hofstadter *et al.*, 1980; Hofstadter, 1985; Mitchell, 1993) If one were to start with the Webster’s dictionary definition of what a “first lady” is, it would be something like, “the wife or hostess of the chief executive of the country” — and if pressed for a definition, probably most people would give something approximating this, although in the United States it would probably be more specific, such as, “the wife of the president of the United States.” As FARG notes, however, many people are still able to come up with an answer to the question, despite the fact that, strictly speaking, there was no wife or hostess of Great Britain’s Prime Minister; namely, Denis Thatcher, Margaret Thatcher’s *husband*. FARG accounts for this ability using the notion of slippage: what happened was that the concept of “wife” in this situation *slipped* to the concept of “husband” in order to find an appropriate possible answer to the question.

This example raises some further issues which highlight more features of FARG’s notion of concepts and how they are structurally related. First, the “dictionary” definition is not supplanted just because a case is found in which a violation of the definition was appropriate. Rather, the definition serves as a “central” or paradigm case for what the concept typically refers to; at the same time, there are relations to other concepts which, under the right conditions, it would be appropriate to slip to. From this feature emerges the view that “central definitions” of concepts are related to one another in specific ways; for example, the concept of “wife” can slip fairly easily to the concept of “husband” because they are related to one another (they are complementary) — but very unlikely to slip to the concept of “cat”, unless very strong pressure was put on that slippage being an appropriate one for the situation. This introduces the notion of a “conceptual metric”: certain concepts are “closer” than other concepts to a given concept. Furthermore, which “direction” the slippage occurs is heavily dependent on context (the mechanism of which will be explained, below).

This structure of the relations of concepts is described by three types of slippages that are possible: export, transport, and import slippage. *Export* slippage occurs in the situation in which a concept that is applicable to a given situation is either made more abstract or “variabilized.” For example, the concept “President” becomes more abstract when it slips to the more general concept category that it belongs to: “chief executive”; likewise, slipping from the concept “cat” to “mammal”, or further, to “animal,” captures the same kind of exporting. Variabilization is similar, except that it involves the situation in which the more general concept requires a variable, which could be filled by any suitable instance of that type. For example, the concept “United States” could undergo an export slippage to the more general category of “Country *X*” — where *X* can be any country. In general, export slippage captures the notion of moving from some subordinate type to its supertype.

The result of an export slippage is an *abstract schema*, which is a definition of the situation according to the abstract (supertype) categories, with possible variables, that were reached through exportation. For example, an export slippage from the representation of a situation according to the concepts “Wife of the President of the United States” becomes the abstract schema “Wife of the chief executive of Country *X*.”

Transport slippage then occurs when an exported abstract schema is adapted to a new concrete framework. Specifically, it involves the rebinding of the free variables in the schema. So, in the above example, “Country *X*” is replaced by “Great Britain.”

Finally, *Import* slippage occurs when an exported and transported slippage has occurred, but the resultant schema still doesn’t apply to the target situation. This inability to “apply” to the schema to the target situation puts pressure on the concept involved in the new export schema that does not fit, and this, in turn, may force a slippage of that concept — specifically a slippage in which that concept is *replaced* by an appropriate, related concept. Thus, in the “First Lady” example, the abstract schema of “Wife of the chief executive of Great Britain” contains a concept that still won’t fit the target situation: namely, the chief executive of Great Britain was female, and also happened to not have a “wife.” However, Margaret Thatcher was married, thus suggesting a possible alternative. The import slippage then had to be made which replaced the concept “wife” with the closely associated concept “husband” (reasonably assumed to be “conceptually close” in that wives and husbands are complementary ideas), which ends up in fact matching with a possible situation because there *is* a “husband of the chief executive” — Denis Thatcher. This third slippage completes the transformation of the “wife of the president of the United States” into “husband of the chief executive of Great Britain.” Because this slippage involves a literal replacement of a concept from the original situation by an associated concept in order to fit the new situation, Hofstadter & Mitchell (1991) refer to this as “true” *conceptual slippage*.

To review, this general overview of the kinds of slippage that can occur within concepts is important because it highlights a number of features which FARG’s ontology of concepts has. Namely, that there are hierarchical relationships between concept categories (again, the example of “cat” as a kind of “mammal,” which in turn is a kind of “animal”), and that there is also an implicit semantic “metric” between concepts (that “wife” and “husband” are closer than “wife” to “car”). And, the notion of flexibility in concepts is captured by the kinds of slippages that might be made.

Codelets and the Parallel Terraced Scan

The “First Lady” example above, however, does not complete the high-level perception story — we do not yet have enough detail to propose an actual “mechanism” for how an analogical agent with such a flexible conceptual structure might make analogies. What are still needed is an account of how perception takes place, and how concepts are wrapped up in that perceptual process. As already noted, Chalmers *et al.* propose that the “end product” of the process of perception is a “representation” of the situation — and this end-product representation will itself comprise the agent’s perspective of the analogical relation that is perceived between to domains in the situation (the source and target). Therefore, the perceptual process is a process in which the representation of the situation is *constructed*. It is in this concept construction

process that the processes analogous to SMT's account of representation structural alignment and eventual mapping (to produce an analogical comparison match) take place.⁶⁰ The crucial difference, of course, is that these processes, for HLP, are wrapped up *in* the process of representation construction, and this construction process involves the concepts of the modelled agent, as described in the above account of slippage and concept relations. More is needed to explain this representation construction.

The general outline for the architecture that pulls these ideas together to account for the construction process is based on the outline in Hofstadter's "Copycat Memo" (Hofstadter, 1984). Here, the basic idea is that there are "low-level" computational units that compete with each other carry out some task. In later models (particularly Copycat and Tabletop), these units are referred to as *codelets*. Each of the codelets does only a specific function, and whether that function is able to be carried out depends on what representational structure is currently present. If a codelet can't "run" because the conditions aren't right, then it "dies" and its place is taken by a new, potentially different kind codelet. If, on the other hand, a codelet *can* run, then it carries out its function, which may range from adding (building) new structure to already present representation, to removing already present structure, to identifying that a certain kind of structure is present, etc.⁶¹ When a successful codelet completes its function, it then spawns a new codelet which does a related function (unlike the unsuccessful codelet, which may be replaced by a number of different kinds of codelets).

This notion of codelet competition became the basis of a series of models (Jumbo: Hofstadter, 1983; Seek-Whence: Meredith, 1986; and Numbo: Defays, 1986) aimed at showing how stochastic competition amongst these codelet functions could demonstrate interesting properties in constructing representations of a situation — a method for simulating a kind of "perception of the situation." Specifically, this stochastic competition was able to construct interesting representations of a situation without specific goals — only a collection of "micro-rules" which stipulated each component's behavior. This idea was inspired by stochastic methods for searching for an "optimal solution to a problem" (in this case, the optimal representation of a situation) — stochastic search methods which were pioneered by the work of John Holland (1992, the second edition to his original 1975 publication).

The basic idea of a stochastic search method is to allow for a number of different possible "solution paths" (different proposals for what might be a good representation of the situation) to be explored at the same time (in a simulated form of parallelism). Initially, all possible paths will get the same amount of computational resources (the

⁶⁰ Although FARG tends to not use the "structural alignment" terminology, and there are important differences between the SMT and HLP algorithms for how alignment and mapping occurs, there is still an important similarity of function between the two: they are *both* accounts of how representation structures (which correspond to features of knowledge about what is represented) come to be matched-up to find how the compared situations are analogous to one another.

⁶¹ There is even the codelet function for proposing the final "rule" for carrying out the function that *completes* the representation construction process — the point at which the program halts. In the HLP models of analogy, this rule is for carrying out the function that is "analogous" to the compared situation, thus completing the analogical comparison process — the resultant structures that were built indicate how the compared situations are analogous to one-another, as described by the final mapping rule function of the last "rule" codelet run.

same potential for the amount of “attention” to possible solution paths), but as certain paths prove to be more “interesting,” computational resources would be focussed on providing more codelets devoted to the kind of representation structure. At the same time, other paths of enquiry would not be completely ruled-out, because the basic proposals for search (decisions about what codelets will be proposed to “run”) will still allow for the possibility of proposing searches in other directions (even if that possibility diminishes with the onset of other, more promising representational structures).

The mechanism devised to account for this measure of “interestingness” of present representational structure, and how that measure will affect further search, is *computational temperature*. Computational temperature is a feedback mechanism which has a direct influence on possibility of codelets running, and therefore plays the integral role in how the path of search “settles” on a particular proposed solution. Initially, the model has a large variety of different codelets, each with their own specific function. Each proposed codelet is assigned an “urgency” value, which determines what its possibility of being “chosen to run” is relative to the other proposed codelets. Codelets that are proposed to run are placed in a data structure, called the *Coderack*. Stochastic-based choice of which codelet will run is then made from what codelets are present on the Coderack. (In the early models, which codelets are selected for the Coderack starts with an initial selection, and then new codelets are added depending on which previously codelets were successful in their run.)

Computational temperature thus affects the over-all selection of codelets by determining “how much” the stochastic selection of codelets is determined by each codelet’s urgency value. Computational temperature, in turn, is affected by what kinds of codelets have been successful in running, and therefore how much present representational structure there is; codelets are rated according to how “complex” or “important” their particular structure-building function is, so that if a number of codelets with a high rating of “importance” have been successful, then this acts as a measure of “how important” current representation structure is. The more “important” structure there is, based on the over-all rating, the “lower the temperature is, which in turn entails that codelets proposed will only be chosen strictly on the basis of their proposed urgency. If, on the other hand, there is little “important” representation structure present, then the temperature is “high”, which in turn causes the stochastic selection of codelets to pay less attention to each codelet’s urgency, and therefore allows for a greater variety of different kinds of codelets to run.

In this sense, the temperature rating can be metaphorically described as rating “how happy” the system is with what has been constructed so far, and exploration of possible promising structures is based on the principle of “exploration proportional to promise”: the more promising the structures present are, the less the need for global exploration (more indiscriminate randomness of search); the less promising they are, the greater the need for global exploration to find other possible answers. The combination of the stochastic choice of codelets, controlled by the global rating of “satisfaction” implemented by the computational temperature feedback mechanism has been labeled the *parallel terraced scan* — “parallel” because a number of possible solutions are scanned at once, and “terraced” because certain codelet functions indicate more promising solution paths and thus more attention will be focussed on the solution paths they indicate when they run. (The behavior of the parallel terraced scan process is like that of

“simulated annealing” (Kirkpatrick *et al.* 1983), except that unlike the fixed time-schedule for the “annealing” process (annealing being a metaphor for the narrowing of the search to a particular path), the “annealing” of the parallel terraced scan works independently of any time schedule and is instead based on the temperature feedback rating.)

The Slipnet

Early models that implemented the parallel terraced scan (Jumbo, Seek-Whence, Numbo) depended entirely on the initial set of codelets, and then what new codelets were chosen (from the whole class of codelets) by the codelet that had just successfully run. This was successful in capturing the general kind of perceptual mechanisms that FARG had been looking for. But, in order to make a model which was based on the notion of “high-level perception,” the parallel terraced scan architecture had to be combined with the model of human concepts that FARG proposed above.

The model of the nature of concepts and their potential for slippage was thus implemented in a semantic network-inspired architecture, called the *Slipnet*. A Slipnet consists of a collection of nodes, each node representing the “core” of a concept that the modelled agent is proposed to have. These concepts could range anywhere from concrete identifications, such as an object like a “cup,” or the name of a role in a relationship, like “wife,” to more abstract relations between concepts, like “opposite of”, etc (these latter concepts are still concepts, and therefore represented as nodes, but they may be linked to the links between two other concepts, in order to capture their possible influence over or activation on the basis of the relationship between the two other concept nodes).

These nodes are then related to each other via a number of different kinds of links (this idea was first robustly modelled by Quillian, 1968). Which nodes are linked to which captures the notion of the “conceptual metric”: how concepts are related to each other, and therefore, what kinds of slippages can occur. The Slipnet links consist of two basic kinds: links expressing class-membership relations (*ISA* or *HAS-MEMBER* links), and *labeled* links expressing the type of relationship that a link between concept nodes is intended to encode.

The first kind of link shows how one concept, e.g. “cup,” is a kind of another concept, “liquid-holder.” This serves to capture how kinds of objects and classes of objects are related: thus a “cup” and “mug” are related in the sense that they are both kinds of “liquid holders.” And the specific relations between a subtype and a supertype is captured by the complementary links *ISA* (a “cup” *is a type of* “liquid holder”) and *HAS-MEMBER* (a “liquid holder” *has a member of* “cup”).

Labeled links, in turn, are comprised of nodes (because they are also concepts about the kind of relations) which are attached to (and therefore may influence or be influenced by) the links between nodes. Because labeled links are themselves concepts, they too may be linked to other concepts. For example, when considering the English alphabet as a linear string, the letter ‘a’ is the predecessor of the letter ‘b’ — thus the label concept (also a node) “predecessor” describes the relationship of ‘a’ to ‘b’. The complementary relationship between ‘b’ to ‘a’ is “successor”: ‘b’ is the successor of ‘a’ — thus, the label concept “successor” describes the relationship of ‘b’ to ‘a’. “Predecessor” and “successor” might then be linked by the labeled link “opposite”, thus construing the

relationship between “predecessor” and “successor” as opposites.

This basic outline of the kinds of relationships captures nicely the “conceptual metric” notion and is useful for a large number of kinds of concepts and their relations. But this in itself does not capture the other important notion of FARG’s view of concepts: slippage. The account for this feature also finds its roots in semantic network research. One of the purposes of Quillian’s (1968) original semantic network model was to capture how certain concepts are related — and therefore how “activation” of one concept could likewise affect the activations of other concepts (“activation” being the metaphor for “what’s currently being thought of” or “what the agent currently takes the situation to be,” even though there is still the presence of other “non-activated” concepts which are not currently playing a role). His network describes this influencing-relation between concepts using the notion of “spreading activation,” in which activation of one node “spreads” to other nodes, thus activating them as well. And the greater the activation of one concept, the more that activation could spread to other nodes, and the nodes they’re related to, and so on; one could model spread of activation over time, or how the “spread” might degrade over time, or number of nodes traveled from the original activated node (a kind of “distance”), etc.

This spreading activation notion is also employed in the Slipnet, except that rather than having activation spread at pre-determined amounts, rates or distances, the labeled links in the Slipnet have a value which describes their *conceptual distance*, and these values can change. Two concepts that are linked by a labeled link are “closer” or “farther” from each other depending on the *size* (a numerical value) of the conceptual distance. Conceptual distance, in turn, is affected by the activation the label associated with that link. Conceptual distance designates “how easy it is” for activation to spread from one activated node to its linked partner — the greater the distance, the harder it is to pass activation; the smaller the distance, the easier to pass activation. Also, activation of a node has degrees, depending on the presence of something to activate it (explained below) — and, if whatever activates the concept node is not currently present, the activation slowly fades. These architectural features together allow for the modelling of potential flexibility and slippage: if a concept node ‘a’ is activated, and the labeled link “predecessor” becomes active, thus making the conceptual distance between ‘a’ and ‘b’ small, then it is much easier for activation to spread to ‘b’, thus making it active. Likewise, there is the potential that ‘a’ and ‘b’ could be active and in turn affect the activation of the labeled concept “predecessor” or “successor” (clearly there are more details to this story, but they’re not important here — only the capacity).

And a final feature of the Slipnet architecture concerns the implicit hierarchical relationship between the labeled concept-nodes. As already mentioned, labeled nodes can themselves have labeled nodes between them, and so on. This allows for a notion of *conceptual depth*: the idea that certain concepts are more abstract (“deeper”) than others. For example, the rather concrete concept of the identity of the letter ‘a’ is not nearly as “abstract” as the concept of “successor” (this is now a relational term that applies to things in a line), which in turn is not as “abstract” as the concept for “opposite” (also a relational term, but has much more applicability; it applies to anything with a complement). In similarity comparison, we typically judge a description of the comparison as “deeper,” “more profound,” or “better” the more abstract the concept involved in the comparison is. This makes sense because the deeper or more abstract the

concept is that is used to describe a situation, the more underlying relations (which are less abstract and related to the concept) there are that will be inherently involved. The Slipnet architecture cashes-out this intuition by assigning pre-established difference in activation-fading which increase the length of time that it takes for the “conceptually deeper” concepts to loose their activation. Because these “conceptually deep” concepts will stay activated longer once they are activated, they have a much greater influence on the concepts linked to them which are “lower” on the hierarchy. Also, activation of concepts with higher-order conceptual depth leads to a relative measure of greater “satisfaction” that the conceptual representation of the situation is interesting (the feedback mechanism which constitutes this “measure of satisfaction” is the computational temperature, described below).

(Note that this amounts to causing the HLP architecture to favor relations between structures because conceptually deeper concepts will have a greater impact on processing than conceptually shallow concepts, and conceptually deep concepts won’t get activated until a requisite amount of representational structure exists — the higher-order the relations, the conceptually-deeper the concept activated. This is roughly the same systematicity principle that SME models, and which SMT has found psychological evidence for: that people tend to favor mapping systematic structures with higher levels of abstraction over structures that are not consistent or involve lower-order relations.)

Putting it all together

With the components for representation construction and the Slipnet model of concepts, the now well-rehearsed insight from Kant comes to the foreground: “concepts without percepts are empty; percepts without concepts are blind.” Chalmers *et al.* in fact note that the HLP architecture is inspired by Kant’s conception of the mind as comprised of two faculties: the faculty of sensibility (which for Chalmers *et al.* corresponds to low-level perception) and the faculty of *understanding* (which for Chalmers *et al.* corresponds to high-level perception) — the involvement of concepts (basic categories) in organizing information that is perceived (that has already been through initial processing of low-level perception). Understanding is what high-level perception is all about; so the Slipnet needs to be hooked up with the operation of the parallel terraced scan method of representation construction.

As Hofstadter (1995) and French (1995) note, the early “perception” models which involved the parallel terraced scan architecture operated, in a certain sense, “blind” to what they were doing because they worked entirely “bottom-up,” with no specific goals orchestrating their behavior (except those implicit in the codelet functions and the associate codelets they activated). By adding an account of high-level concepts about what exists in the world (anything from concrete perceptual categories to higher-level abstract relations), and therefore *an idea of what is interesting in the world*, the search for analogies could now be guided by both bottom-up *and* top-down search: bottom-up search via a competition of construction and destruction of structured representations of the situation, and top-down influence of concepts in the sense of how current structured representations which are built might be taken to represent interesting relationships

(specifically, interesting relationships in similarity comparisons) in the world.⁶²

This mutual top-down and bottom-up activity is achieved by having the activation of concepts affect codelet activity, and codelet activity influence concepts — both top-down and bottom-up influences occurring in parallel. Top-down influence is achieved by having activation of concepts affect the “urgency” values of codelets, thus influencing which codelets have a greater chance of being selected to run. At the same time, bottom-up influence is achieved by having certain codelet behaviors (either in that codelet successfully running, or in the presence of representational structure that was built by a codelet and still endures) raise the activation levels of corresponding Slipnet concept-nodes. For example, a certain codelet which has the task of indicating that there is a “successor” relationship in the current representation structure would increase activation in the “successor” concept-node in the Slipnet; and as long as that “successor” relationship persists, the “successor” concept will stay activated; and if more “successor” relationships are indicated, then the “successor” concept will be activated even more. Activation would likewise spread in the Slipnet as a result of the “successor” concept-node’s activation, as described above. This, in turn, could lead to the further top-down influence: the activation of the “successor” concept-node will, for example, raise the urgency level of codelets related to identifying “successor” relationships (such as “successor” perceivers). And so on.

A final architectural component to add to this integrated architecture is the “place” where representation construction takes place: this is called the *Workspace*.⁶³ The initial Workspace starts with the raw input situation. The initial raw input is made up of component parts of which must have a least one codelet type that can identify them as being a certain kind (otherwise the HLP mechanism would be “blind” to those aspects of the raw input). As new codelets run and add representational structure, destroy structure, perceive structure (activating certain Slipnet concepts), or propose mapping rules, the current state of the representation construction process is kept in the Workspace. This use made of the Workspace falls under the class of computational models called “Blackboard” systems; the intended allusion being the subsequent additions or subtractions of figures drawn on a chalk blackboard, where at any given time one could observe the current state of the blackboard and see the results of what processes (minus structures that had been erased) led up to the current state.

Computational temperature also functions roughly the same way as in the simple parallel terraced scan models, although now it is a complex function of the current Workspace representation-structure state and the activation level of the Slipnet (activation of “conceptually deeper” concepts leading to “greater satisfaction” with operation — and therefore, lower temperature).

Figure 3.10 depicts the four main components of the HLP architecture and their interrelations:

⁶² Note: it would be interesting to account for how the agent comes to find certain relationships as interesting for similarity comparisons.

⁶³ Hofstadter (1984) originally referred to the Workspace “Cytoplasm”, employing a descriptive metaphorical reference to the activity of many enzymes building and breaking structures simultaneously throughout the cytoplasm of a cell.

(which “lives” in the domain) might be presented with: the situations which the agent is initially presented with in its Workspace, and with which the agent is to then find an analogical relationship.

Copycat, for example, operates in the “letter-string” domain. The letter-string domain is defined as follows:

- (1) It is comprised of the 26 letters of the English alphabet, from ‘a’ to ‘z’. Each of these entities has an identity; hence an instance of ‘a’ is the same as another instance of ‘a’, and not the same as an instance of ‘b’, etc.)
- (2) The alphabet is arranged in sequential order (‘a’ is adjacent to ‘b’, ‘b’ is adjacent to ‘c’, etc.);
- (3) There is a sense of “direction” in the sequential arrangement (‘a’ is the predecessor of ‘b’, and ‘b’ is the successor of ‘a’);
- (4) The alphabet is “non-circular” (there is no predecessor of ‘a’ and no successor of ‘z’ — ‘a’ and ‘z’ are “endpoints” of the letter-domain)

Based on this underlying ontology of entities and their possible relations, Copycat (the modelled cognitive agent) is then presented with a “problem situation” whose semantics is based on this underlying letter-domain ontology. The general scheme of the “problem situation” Copycat is presented with is a *partial analogy-square* consisting of related strings of letters:

Figure 3.11 - Partial Analogy-Square

The *letter-strings* in Figure 3.11 are small groups of letters from the alphabet domain arranged in various orders (e.g., **abc** __ **abd** as **hij** _ ?). Copycat’s task is to find the general rule that transforms letter-string 1 into letter-string 2 (the source), and then apply that transformation rule to analogously change (following the same general rule) letter-string 3 into the yet-to-be-determined letter-string 4 (the target). This is Copycat’s basic task in the domain.⁶⁵

⁶⁵ Tabletop has a very different kind of “problem situation”: in the Tabletop domain, there are two “people” sitting across from each other at a table. On the table are a number of different kinds of objects (typical things found on a table, ranging from cutlery to dishes and various kinds of liquid holding objects, etc.) in different spacial arrangements — in general, there is a different group of arrangements in front of each person. The two people are Tabletop (the modelled agent) and the “problem presenter.” The problem presenter points to one of the objects on the table and says, “Do this!” It is then up to Tabletop to perceive what pointing action would be analogous to what the problem presenter did.

While this definition of the domain and the problems that may appear in it suggests that Copycat is somehow “independent” of the domain it works in or the problem situation presented to it, this is not entirely true. The relationship between the domain and the specifics of Copycat’s architecture is more complex. Certainly, there is going to be a wide variety of different ways in which a presented problem might be perceived. But, Copycat must antecedently have some way of recognizing the identity of the letters in the domain, as well as any of the possible relationships which it might find between those identified, otherwise Copycat would be, as already mentioned at the end of Section 5.4 above, “blind” to these entities or their relationships. Copycat’s representation construction powers (= Copycat’s perceptual powers) are found in the combination of both (1) the codelets, whose simple functions are oriented towards identifying the letters and other categories and their possible relationships in the presented problem situation, and (2) the Slipnet concepts, in which the categorical identifications of what kinds of letters, relations, and relationships between the strings in the problem presentation there might be.

Copycat, in fact, is given perceptual capabilities which go beyond just the domain definition of what entities exist and how they are presented. For example, Copycat is also given some additional perceptual/representational power in that it is capable of recognizing groups of letters (e.g., if letter-string 1 is comprised of **aaabbc**, Copycat has the capacity to possibly represent the first three letters as a “group of *same letters*” — or additionally, a group of **a**’s), and also has rudimentary counting capacity (in the previous example, the letter-string has a group of 3 letters, followed by a group of 2 letters, and then 1 letter; Copycat can “count” up to 5). In addition to these perceptual enhancements, Copycat also has some higher-order representation capacity, such as understanding that successor relationships are the *opposite* of predecessor relationships, and this notion of opposite can also help in Copycat’s search for the transformation rule that changes letter-string 1 to letter-string 2, and will eventually be employed to analogously change letter-string 3 in to a proposed letter-string 4.

In Copycat, there are a total of six kinds of representational structure of the problem situation that can be represented in the Workspace, each type built by a corresponding codelet function:

- (1) *descriptions* of objects (e.g., *leftmost* as a description of the **a** in **abc**),
- (2) *bonds* representing relations between objects in the same string (e.g., a successorship bond between the **a** and the **b** in **abc**),
- (3) *groups* of objects in the same string (e.g., the **ii** group in **ijjkk**),
- (4) *correspondences* between objects in different strings (e.g., a **c-kk** correspondence in **abc _ abd, ijjkk _ ?**),
- (5) a *rule* describing the change from the initial (letter-string 1) to the modified string (letter-string 2); For example, “Replace the rightmost letter by its successor”,
- (6) a *translated rule* describing how the target string should be modified to produce

an answer string (e.g., “Replace the rightmost *group* by its successor”). When this codelet runs, Copycat is finished in making its analogical comparison because the translated rule carries out the final “mapping” on the basis of already present structure to produce the new structure in the target (the transformation of letter-string 3 into the proposed structure of the previously vacant letter-string 4) that is analogous to the source (the transformation rule of letter-strings 1&2).

Figure 3.12 - Representation structures in the Workspace of Copycat
(a) initial problem presentation; (b) representational structure at a later time

Figure 3.12 is a sample contrast of an initial “problem situation” presented to Copycat, and an example of the kinds of structural relationships graphically represented in the Workspace later-on in processing. And the slipnet likewise has the concepts which correspond to the different sorts of relationship categories that could be found and guides the production of new codelets related to these categories on the basis of which categories are activated in the Slipnet.

In sum, all of these perceptual capacities (i.e., the ability to construct certain representations on the basis of the presented situation) in the current HLP models (Copycat and Tabletop) are a result of the combination of (1) codelet types and their functions, (2) Slipnet concept-node types (and their functional relations), and (3) the bottom-up / top-down influencing relationships between the activity of these codelets and the Slipnet nodes. The combination of these three functional components in the process of representation construction define a “space” of possible representations of the situation that can be built. Certainly, there is quite a bit of flexibility as to what representations will be constructed on the basis of a particular problem situation presentation. Therefore, it is best to think of the codelet functions plus Slipnet nodes and relations, in addition to the bottom-up / top-down activation influences between the codelets and Slipnet as defining a kind of *representational grammar*: a “grammar” which stipulates what kinds of representational structure can be constructed (and therefore “perceived”). Describing

this as a representational *grammar* is appropriate because, just as we would expect with any grammar, the representational grammar of the HLP models determines the “rules of correctness” for application of certain kinds of representation structure construction processes (including “destruction of structure” and “rule proposals”)⁶⁶: that is, a stipulation of what can be built (representational capacity)⁶⁷ based on what is present in the Workspace, as well as what cannot be built (representational constraints)⁶⁸. And, certainly, this representation construction grammar, like any grammar, is also *productive* in the sense that many different kinds of combinations and permutations of codelet representation structure construction is possible — provided that the rules stipulating the conditions under which a codelet can run are not violated (i.e., you don’t break the rules of the grammar). This notion of a representation grammar is very important for the evaluation of HLP’s representation capacities, as I will discuss below in Section 6.

(In an important sense, the “domain” which Copycat and the other HLP models work in is explicitly represented *in* the modelled agent’s Slipnet and codelet representation construction grammar. I will discuss in Section 6 what impact this has on SMT’s account of representation with respect to representation emergence.)

One of the immediately obvious interesting differences between the SME and HLP - based approaches to representation is that in the SME-based models, while there are structures of relations which have an explicit account of different kinds of levels of represented relations, there is not the same kind of “depth” of different *roles* which representations play in the structure alignment and mapping processes as found in the HLP models. In the HLP models, the role of representational dynamics spans from a kind of “short-term memory,” where an “actual representation of the situation” is constructed, to the deeper role of represented concepts in a kind of “long-term memory.” The key distinction between the two levels is that there are not just representations that have dynamics with respect to compared structures on the basis of context, but there is also a graded role of representational dynamics, ranging from complete construction of associations in the “short-term memory” (the Workspace), to more “stable” representational roles as locations in the Slipnet “long-term memory,” which also still have flexibility (such as changeable conceptual distance and activation levels).

Certainly, the HLP models have introduced to the story of the explanation of analogy an interesting potential account of how dynamics of representation construction itself can be implicated in the alignment and mapping of representational structure in the process of an analogical comparison. But whether this account has avoided handcoding that restricts an account of representational content and emergence is yet to be determined.

⁶⁶ There is, of course, an important difference in how the rules of the grammar are “followed” — namely, the difference between normative rule-following and causal rule-following (Wittgenstein, 1953). In this case, the grammar describes causal rule-following.

⁶⁷ For example, that a codelet for the recognition of the letter **a** in a presented situation in the Workspace can “run” (if selected to run) and therefore construct the structure of the identification of the letter-type *a*, which in turn initiates activation of the concept-node ‘a’ in the Slipnet.

⁶⁸ For example, the codelet which has the function of recognizing an instance of the letter **a** in the presented situation of the Workspace *cannot* have the function to build a structure which indicates an “opposite” relationship (that is not its designed function); nor could this codelet “run” on the basis of the letter **c** being present — it’s only designed to be able to run in the presence of the letter **a**.

6 - Handcoding of representation in the SMT and HLP models

I have reviewed the central SMT and HLP models, their representational assumptions, and how they make use of those representations. I will now consider how they are handcoded with respect to representation emergence and change. A review of my constraints will re-emphasize why it is important to consider whether and how such handcoding has occurred: My eventual goal (beyond the scope of this dissertation) is to be able to account for how new representational content can *emerge* or existing representational content can *change* as a result of analogical processing — that is, I want an account of the mechanism for such emergence and/or change as it is implicated in analogy processing. But, I don't want to *handcode* the emergence or change of representational content. That is, I do *not* want a model in which either of the following are *necessary*:

- (1) A human researcher that necessarily has to put a new label or functional unit in the model when emergence or change is required — since the human researcher is something whose inner-workings we do not have direct access to, necessary reliance on the human for the creation of new representation entails that we still lack an explanation of how that representation could emerge or change.
- (2) A mechanism for emergence or change in the model that has to “spawn” some “new-representation-marker” which it is then up to the observer of the system to interpret. Again, if the meaning of the representation has to be applied by an *already* representing observer (who's representing power we have yet to explain), then we still lack an explanation of how the meaning gets there, or what meaning there *is*.

6.1 - “Content” ... and determining it in computational models

In order to determine whether the current SMT and HLP models can offer a non-handcoded account of representation emergence or change, I need to first have some way of characterizing what it is that is to emerge and change. I start with the following general assumption: that these internal representations are intended to characterize what the agent knows.⁶⁹ That is, I take it that what is construed as the “agent's representation” is a metaphorical way of talking about what the agent knows about its world from its psychological perspective (i.e., how the agent conceives of its world — I do not mean “knows” in the sense of “justified true belief,” nor does this depend on any notion of

⁶⁹ Knowing and knowledge, of course, are quite complicated concepts in their own right. At this point I'm using the phrase, “what the agent knows,” as potentially applicable to all kinds of knowing, from “skill knowledge” or “knowing how” to “knowing that,” etc. More specific distinctions and commitments will be made in Chapter 4.

consciousness). And determining what it is that these representations provide the agent (as the agent's knowledge based on having or making use of these representations) is based on what the *content* of a representation is.

Therefore, what I need is an objective way of determining what the content *is* of the representations of these models.⁷⁰ By “objective way of determining content,” I mean having some kind of criteria which serve to make a consistent interpretation of the content the modelled system has — something about how the *system* is set-up — and not based solely on the particular interpretation an observer might apply. These criteria will be based on features of the model.

Lacking some objective account of what content there is in these representations means necessarily relying on interpretation by an observer, which, if we are interested in accounting for the emergence or change of content, amounts to requiring handcoding in order for content to be determined — and this requirement of handcoding, in turn, entails a lack of an explanation of content emergence or change. It follows from this that if it is not possible to interpret a model's content according to some independent criteria (i.e., independent of necessary interpretation by an already representing observer), then the necessity of an interpreting, semantics-attributing human cannot be avoided. For my objectives here (a non-handcoded account of emergence and change), this is unacceptable.

6.2 - Wishful Mnemonics and The Eliza effect

Given these assumptions, how do I begin characterizing this content with observer-independent criteria? I am going to pick up on the early observation of Drew McDermott (1976) that there has been, and continues to be, an unfortunately damaging practice (particularly for the efforts of AI and cognitive science, which are aimed at descriptive and explanatory models, not just engineering achievements) of labeling entities or functions in programs with fancy linguistic titles that suggest more than they deliver — what McDermott calls “wishful mnemonics.” In addition to McDermott, Weizenbaum (1976), Boden (1977) and most recently, Hofstadter & FARG (1995), have noted that these labels can be quite compelling to interpreters of the programs, especially as the complexity of the program increases. Hofstadter uses the colorful term *the Eliza effect*⁷¹ to refer to this human interpreter susceptibility. Hofstadter & FARG (1995, p.157) defines the Eliza effect as, “the susceptibility of people to read far more understanding than is warranted into the strings of symbols — especially words,” attached to or associated with the entities or functions in programs espoused as models of how aspects of cognition work.

⁷⁰ I am starting at this level of “objective” interpretation, but the real crux of the problem is actually a deep metaphysical issue concerning the *ontological* status and nature of cognitive representation in general — the metaphysical issue is more primary than the interpretational issue; or, better, how we ultimately interpret representation will follow from our ontological commitments. This foundation will be fully exposed and explored in Chapter 4.

⁷¹ ... named after Wizenbaum's (1966) “canned-response” computer simulation of a psychotherapist, name ELIZA; I discussed ELIZA in Chapter 1.

While “wishful mnemonics” may cause interpreters of a program’s capabilities to fall prey to the Eliza effect when considering fancy procedures (e.g., an “understanding loop” that is more accurately described as a “node-net-intersection-finder” procedure), my concern regards what happens when it comes to interpreting the representational content of proposed representations. The mistaken tendency in this case would be to label some entity in a program with a linguistic title, for example ‘cat’, and then assume that *because the entity has that label*, then it must be *about* or have the content about cats (or some cat) in the way that we have some representation in our brain of cats when we are presented with (perceive) or think about cats. The concern here is that such attributions could lead to falsely attributing content or semantics that isn’t really there (in the model as constructed by its author) to the tokens or objects intended to be playing a representational role.

The attribution of false content has particularly negative consequences for any representation-based explanation⁷² of cognition that is based on the performance of a computational model. The explanations that these models are intended to provide are based on the “picture” we get when we observe the processes that operate on, manipulate, combine and re-arrange these labeled representations. The methodology shared by the two analogy research programmes I have investigated posits that after these processes are completed, the new organization of representations is intended to demonstrate how or whether the modelled agent has accomplished some cognitive feat (e.g., making an analogical comparison). And our ability to determine what that outcome is, is based on how we interpret the contents of the newly arranged, resultant representations. Ideally, this interpretation should be based on criteria *independent* of (i.e., not solely dependent upon) our own ability to understand certain linguistic labels as being about the things we take them to be in the same way as when we read and comprehend words in other contexts (like reading this paper, or a book). We want these criteria to be such that they determine our interpretation not by our normal language-comprehension ability (e.g., we read the word “cats” and think of cats), but instead dependent on how the computational model is set up.

The consequence of *falsely* attributing content is that if we are mistaken about our attribution of content to these representations, then we are misconstruing what the computational model actually demonstrates. This, in turn, means that the model may be

⁷² Note: currently, a “representation-based explanation” entails an explanation of a phenomena from “within” only the system which makes use of the representation (e.g., inside the head of the cognitive agent); my eventual move to interactivism will entail a shift of the “location” of explanation to phenomena within the epistemic system (or subsystem) *and* what the system interacts with — *and* will be a matter of the history of interaction between the system and its environment (even if this environment is still within the agent), over time. Hence, this move entails a shift to a *situated* explanation (hence, the title, *Situated Representation*). This will *not* be a denial of the possibility of examining the “inside” story, but demonstrates that explaining representation is both an “inside” and “outside” story (of a representing system), and involves the relation between the two. So, there is still an “inside-the-cognitive-agent” story that is crucial, but it is one now situated in a history of development through interaction with the environment (again, this environment is the environment local to the system or subsystem making use of a representation — this local environment could still be *within* the agent, as in mental modelling or imagination). And without this *situation* (this context in which the “inside story” is set, and how it makes contact with this context), we can’t fully address certain representation issues — like, e.g., content emergence and change.

interpreted as predicting the “same” kind of outcome as we observe in the naturally occurring phenomena, when in fact the model’s description of underlying processes may not actually work at all like the phenomena it is intended to model.

Of course, we don’t want to go to the point of saying that *just because there is* a label in a computational model that this false-attribution mistake has been made. (After all, I’m not advocating giving up computational modeling, the medium of which is based on tokened computational objects and processes.⁷³) Instead, what we need is a technique for grounding our attribution of content — some way of steering ourselves away from possible false content attribution.

In order to avoid falling prey to the Eliza effect, I propose the following technique (a version of that used by Hofstadter, 1995): *consistently* remove the labels associated (i.e., attached to the proposed representational units or processes in the computational model) or replace them with other names, and then judge what it seems the semantics or content of the representations (or representational processes) are before, during and after model operation. By “consistent replacement” I mean that if a representation element with the label “cat” is changed to “sun,” then every other representation element labeled “cat” must also be changed to “sun.” “Consistent removal” likewise means to remove all the instances of a representation element label. There are, however, some further details to add to this technique, which I will address below.

As Hofstadter notes, this technique can be very helpful in deflating mistaken assumptions about the content of a representation solely on the basis of what it’s labeled, and it helps us to focus on what actual work such content is doing in the operation of the model. This point bears two additional remarks about this technique that should be emphasized. First, this technique suggests that the most concrete way to establish what content is in the model is to consider what role the content plays in further system processing in the model — that is, what kind of “work” such content is doing in the operation of the model. Thus, if the label change (or change of the labeled unit’s functional role) has an affect on further system processing, then we can go about characterizing what that content’s role *is* based on what that change of processing is. This, then, gives us some independent criteria for determining how content is characterized for that model — that is, the determination of the content will be grounded

⁷³ As I concluded in Chapter 2, I am working within the framework of minimal empirical computationalism, which utilizes a framework for describing (interpreting) physical systems as computing functions (given an appropriate mapping based on a measuring theory which consistently maps an SVM to kinds of physical state changes of the system being modelled). But such description (and the fact that it is computational), in and of itself, is *prior* to a theory of cognitive representation — we may be able to describe cognitive representation in terms of computation, but the computational description itself does not assume this. Following this, computational entities, states and process in an SVM are not assumed or required to be representational or play some representational role. Instead, they may do so if organized in a certain way or are part of a particular relationship; what this organization or relationship or process may be, in turn, depends on our (separate) theory of representation. Of course, whether minimal empirical computationalism is true depends on whether cognition (and therefore cognitive representation) can be describable computationally — that is, minimal computationalism adds the additional claims that SVM computational descriptions are sufficient (and the best) for explaining cognition. The point of making this explicit is that some approaches taken to computation assume that computation *is* over representations or representational units. As will be clear in Chapter 4, I take the position that there are computations, or physical systems which have SVM descriptions, which do *not* involve representation.

in the *system's processing*, not just on how we interpret the relation of labels to what we intuitively think of those labels as being about. (Here, the guiding principle for the move to make content a matter of what it does in system processing is to capture the sense that our cognitive functioning depends on what it is we think about — what we represent. That is, that we think, draw inferences and potentially behave differently depending on whether we are thinking about the weather or about what we had for breakfast; certainly the former is more likely to remind us to bring our umbrella than the latter.)

The second point is that this replacement is to be made with the “representational elements” of the model. That is, we are replacing elements that are posited as playing a *representational role* in the cognitive agent that is being modelled. As we shall see, from this perspective the replacement/change-of-label tool exposes that in the models I have surveyed thus far there are two general classes of approaches to the role of proposed representational content in model operation. The first class, *Class 1*, is characterized as modelling in which the entire model is composed of the cognitive agent (or part of the agent) being modelled, with no additional domain or environment proposed to which that model is representationally related (i.e., a domain or environment which the representations are proposed to be *about*). This is not to say that such models aren't intended to be interpreted as being in the same world that we are, but that the computational model itself does *not* attempt to also model an environment or domain the agent is in, *or* how the agent interfaces with some environment or domain — the model is intended to only model what is “in the head” of the agent. For this reason, any replacement/change that goes on with this class of models will be consistent across *all* of the representational structures in the model, since no distinction is made between what is “internal” to the agent (what is serving as a representation) as opposed to what is “external” to the agent (what might be represented) — it is all part of the same internal world of representations inside the agent. This first class includes the basic SME model and two of its extensions, I-SME and MAC/FAC. I will collectively refer to the models in Class 1 as ‘SME’ models for the reason that SME's treatment of representation is the underlying inspiration for that found in I-SME and MAC/FAC.

Class 2 models are characterized as models in which there *is* a modelled account of a domain or environment which the “cognitive agent” being modelled is distinct from. Identification of these models as having made this distinction is not made on the basis of whatever background assumptions there may be concerning whether these models are to be interpreted as being in a world. Rather, this class of models specifically *has* a computational account, *within* the models, of a relationship with some environment or domain — in these models, there is a distinction between the cognitive agent being modelled and the domain. Here, only the structures of the model that are “within” the cognitive agent play a representational role — and thus, the replacement or change tool is to only be used with these “internal” structures. (I should also note that there are further distinctions to be made within this second class: e.g., to what extent the domain or environment is independent of the modelled agent, and what their relationship is with their domain or environment. I will discuss these, below, through specific examples). This second class includes the HLP models of Copycat and Tabletop. (This class also includes some important extensions of the SME-based model family, such as Phineas; however, I will discuss and evaluate them in Section 7, after presenting the basic handcoding critique.)

6.3 - Handcoding in Class 1: SME's representations

Consider a possible comparison between two simple sets of source-target pairs that might be given to SME (Figure 3.13) — here I have taken the representations of comparison A and then consistently replaced the labeled contents to arrive at the comparison B situation:

Figure 3.13 - Changing labels in an SME-style representation

Even though the labeled contents of the comparison A and comparison B situations are very different (particularly with respect to the situations which the labeled contents suggest), the comparisons would be processed exactly the same way by SME, regardless of the “content” suggested by each set’s labeled representation (assuming that we’ve replaced content-labels consistently). In fact, as the general form to the right of the two comparisons in Figure 3.13 shows, we could arbitrarily pick any consistent replacement for aRb ⁷⁴, p , q , r and s and get exactly the same functioning of the alignment and mapping process. And this generalizes to whether new representations are added during the mapping process (still assuming consistent replacement), as I-SME models, or whether there is first a “surface” comparison with a large database of structures before choosing candidate structures for mapping (also assuming consistent replacement in the database), as MAC/FAC models.

On the other hand, if a source or target from group A were compared with a source or target from group B, the comparison would not work — here, the label-differences *do* matter. What this drives home is that SME’s processing relies only on the identity of

⁷⁴ ‘ aRb ’ symbolizes a relation with two arguments: “ a is related to b ” in some way.

names on the nodes (i.e., that “revolves-around” is the same as “revolves-around” and not the same as “sits-on”), and it therefore doesn’t matter what the specific content of the names are as long as they’re either the same or different. In this sense, the specifics of content beyond identity between two compared contents “drops out” as important to the processing of the system.

Aside from issues that others have raised as potential problems with this approach to

representation⁷⁵, it is clear that an account of fundamental representation emergence and change is *not* available here. First, if any new representation primitive is to be added to the representations in SME, it has to be given to it by an external source: namely, a human (or some other, yet unspecified mechanism) that handles attributing content to the label on the node. And second, even if it were to be added, the content would only have

⁷⁵ I'll briefly summarize three of the major criticisms and some of SMT's rebuttals that have developed over the last five to ten years; I have already mentioned the first two criticisms and their rebuttals (in Section 5.2 of this Chapter), but summarizing them again here is relevant in order to pull these points together, and to relate and contrast them to a third criticism, which I have not before mentioned.

Chalmers *et al.* (1992), French (1995), Hofstadter (1995), and Mitchell (1993) present two of these general attacks: (1) These representations are too anemic to be realistic compared to the kinds of representations that must be involved when we make such high-level comparisons. (2) This can't be a robust model of analogy if the key analogy mechanism does not even pay attention to what the representation is about: SME's representational scheme requires that the researcher antecedently pick out the structure of the representations to be compared — but this is most of the analogy work because all that is left is finding maximal relational identity matches. This latter attack (as described in Section 5.2) is clearly a claim that a kind of handcoding is being committed: in this case, with respect to researcher involvement in setting up the problem. It is argued that the researcher has already handcoded the possible analogies by doing this set-up; I will make a related point, below, but from a distinctly different direction.

As also mentioned in Section 5.2, SMT and its family of models do have some replies to these attacks (Forbus *et al.*, in press). First of all, these are just the “psychological representations” of an agent, so the agent modelled may only “know” about what appears in the representational relations; the above criticisms are therefore about the context of developmental history, not SME or analogy in particular. Of course, there appears to be a problem in that it seems strange to have such a simple representing agent able to handle concepts like “revolves-around”, “electrons”, etc. However, SMT rightly points out (Markman, personal communication, 1997) that it is an open question as to whether we in fact *do* represent, at a high-level, situations such as this in a relatively simple manner — if one were pressed for more detail, the story goes, then more associated information could be recalled to fill in more detail in the mapping process. In fact, SMT offers more details to a story like this with the notion of “unpacking” concepts over developmental time (Kotovsky & Gentner, 1990; Gentner *et al.*, 1995); and MAC/FAC is a model of such a possible background corpus of further representational information (Forbus, Gentner & Law, 1995), and I-SME accounts for how mappings can change with the introduction of new information that might be brought into the mapping process (Forbus, Ferguson & Gentner, 1994). This debate, however, is far from being settled (e.g., if there are supposed to be associations to underlying information, *how* do those associations work?; And why can't they play a role in the mapping process?).

The third criticism involves two related points, and has a similarity to the FARG critiques, but with a somewhat different emphasis (one that FARG would probably also agree to): First, Way (1991) argues that metaphors [and analogies] involve comparisons or mappings of representations of situations that entail whole background theories — and these theories concern what the representation of the situation could be; SMT does not account for much in the way of a robust background-theory structure which has entailments for global structure of knowledge. Way refers to such a structure as a *dynamic type-hierarchy* representation of knowledge (the Slipnet form of representation does incorporate a number of attributes which function like a type-hierarchy). Second, Way also points out that it seems implausible that having such a strong identity requirement accounts for the mutability and subtlety of meanings that change as a result of the analogy and metaphor comparison processes (an argument for change of both source *and* target as a result of mapping has also been made by Dietrich *et al.*, 1996, and Dietrich, in press; Hofstadter, 1995, likewise argues this point in terms of the necessity of flexibility in perception of the situation). This second point is related to first point regarding type-hierarchies in that strict identity becomes supplanted by a contextual identity theory when taken in the context of a type-hierarchy/background-theory structure; to my knowledge, SMT does not yet address this issue, and doing so would entail backing-off from the strict identity thesis (at least, the strict identity role in SME would then be contextualized identity in the context of the knowledge system SME is embedded in; still, SMT makes no such claim).

the consequence of affecting processing on the basis of being identical or non-identical to other representation labels; this effect is not powerful enough to capture the notion of new content emergence or change — that is, of *what* has emerged or changed.

This second point requires more discussion and an example. To get an idea of *what* might emerge or change, consider the following situation: Suppose that, as the result of an analogical comparison, or *during* the comparison process, the cognitive agent were to discover that there *is* such a thing as a “revolves-around” relationship (even if the agent doesn’t have a linguistic label for it) — a relationship which it didn’t know prior to the analogy.

Figure 3.14 - Different revolves-around situations

For example, suppose the agent knew what a circle was and observed that some things move in paths that are circular, around other objects. Suppose further that the agent then observed a number of different situations in which there were these circular paths traveled, but there were a variety of *different* objects going in such paths and a variety of *different* objects at the center of such paths — the circular paths of the traveling objects also having *different* distances from the objects at the center of the circular path and also traveling in *different* directions (Figure 3.14). Yet, despite these differences, the agent notices the situations are all somehow similar (or, maybe the agent doesn’t even initially see them as similar until the following): The agent takes the further step to then realize, “hey, it’s not just that there are circular paths being travelled that makes these similar, it’s that there’s a general kind of ‘revolving-around’ going on” [this may be just a pictorial concept, without the linguistic label] “— and furthermore, this ‘revolving-around’ means that, despite a number of differences between the situations (e.g., differences between the objects travelling in the circles, differences in the objects at the center, and even differences in the size of the circles or direction of the motion), what is important is that all of the components of these situations play a *kind* of role (a relational role): thing *revolving-around*, thing being *revolved-around*, and a *kind* of revolving (circular).”

This, I argue, is a plausible real case of discovery based on analogical processes, involving the similarity comparison process of alignment and mapping, but *also* crucially involving the production of *new* representational content (a new relation is discovered: revolves-around *and* the abstract roles played in such a relationship) *and* fundamental representational change (situations of circular travel around an object are re-organized and re-categorized [or the potential for future such re-organization of previously learned

concepts and observed situations of circular motion] according this new kind of relation). This amounts to the emergence and change of fundamental representational content — that is, not just the transfer of already existing relations in one situation to another on the basis of identity or non-identity in comparison, but the emergence of a fundamentally new way of representing the world (of parsing it up and relating it), as well as potential fundamental changes to previous ways of representing the world — i.e., a content with *new* identity. This emergence and change of representational content is *not* accounted for in the SME model — nor *can* it be accounted for unless a human (or other unknown computational mechanism) adds a new relation with appropriate label (which in the long run is still not an account because we don't yet know how the human did it, or how this possible computational mechanism works). The problem faced in accomplishing this feat of abstraction or super-category emergence is generally referred to as the *abstraction problem* (Dietrich & Beyer, 1998).

There are a couple of important observations and concerns that might be voiced regarding this example. First, this “discovery” is rather profound, and therefore likely to be rare. And second (related to the first), there is also a rich social context in which typical human learners are embedded, and in this context, words are provided to label situations that the analogical learner is observing — and these situations may even be explicitly pointed out as examples of this new kind of category. I agree entirely with both of these points. However, they in no way detract from the point that the example makes: I argue that such internal emergence and learning (or discovery) of a new representational type and/or change in current representing capacity must still be *possible*. Yes, this example is rather profound given where the agent would have to be starting from; but this discovery, I argue, is still a possibility, and something is wrong if we can't, in principle, account for this possibility (i.e., without having to necessarily rely on handcoding).

Also, even in “tutoring” situations, in which a more competent representor and language-user uses a word in a situation to specifically pick out a state of affairs and how it is to be described and conceived of, the learner *must* still be able to make this kind of representational leap or extension in order to be able to use these new descriptive words competently to refer to the kind of situation the tutor intends the words to be used for. That is, the agent must still come to *represent* the world in this new way (learn that there is such a relationship and this is how we refer to it). I also argue that the learner could not learn the proper criteria for application of the concept unless it could make (or comprehend) the analogy on the basis of the similarities and differences between the situations being presented — so the analogy-mechanism is also a necessary component of this learning-in-the-presence-of-a-tutor possibility (this makes analogy-making a central feature of learning). Furthermore, such extension and change is still *necessarily* at work *even* if the tutor is having to shape the way the learner uses the word (concept) as the learner makes small mistakes and is corrected by the tutor. All of this necessarily involves the kind of representational flexibility that SME's current approach to representation just doesn't allow.

The current lack of such a mechanism, of course, does not in itself spell doom for the general representational approach that SME assumes; much more work is needed in characterizing the specific properties of SME's approach to representation and demonstrating the approach's deep problems before it can be concluded that it is *in*

principle impossible for an account of emergence and change to be made (this is the subject of Chapter 4). However, at this first pass, it does not seem possible to make such a mechanism independent of having something (some other computational mechanism or human) pose a labeled object and somehow manage its content so that interpretation by an outside human (or other representor) is not necessary for there to be the contents like those which the labels suggest.

At this point, what seems to make the SME-based models prone to this dependency on handcoding is that the issues of *where* this content might come from or how it might be grounded (i.e., independently determined) are not addressed. And this seems to be indicative of Class 1 models in general: a lack of a modelling account of the separation between the cognitive agent and a domain or environment, and a lack of a computational account of what goes into the cognitive agent's processes for representing that domain. Rather, the labeled content is simply assumed to be there, by fiat, with the assumption that it captures what it is a human mind does in representing (at least the aspect of representation which is employed in a similarity comparison, which, again, is all that SME is modelling), or that the representations can unproblematically be assumed to correspond to situations in the world.

6.4 - Handcoding in Class 2: high-level perception approaches to representation

Just as with the Class 1 models, the extent of the role of representation and “content” in those representations in Class 2 models is made clear by again utilizing the representation replacement/change technique. There are two models which I have reviewed that fall into the Class 2 category: Copycat and Tabletop. Class 2 models pose an interesting contrast to those found in Class 1 in that some new constraints on the “label-replacement” technique must be introduced so that the technique can be used to independently determine the role of “content” of the representations in Class 2 models. At the same time, I still want to maintain the original intent of the label-replacement technique, which is to keep us from being misled by how we read the linguistic labels put on parts of the model that are claimed to play a representational role.

Figure 3.15 - Class 2 models

The first difference is that in Class 2 models there is an explicit account of a domain in which the modelled cognitive agent is embedded (Figure 3.15): Copycat operates within the letter-domain and Tabletop operates within the table-domain (of spatially arranged cutlery & tableware). Each model is presented with a *situation* based on the

definition of the domain; the components that the cognitive agent is to make an analogical comparison between are found in the presented situation.

These domains, in turn, are explicitly accounted for *in* each model in the sense that the internal representations of the cognitive agent function according to the situation (based on the domain definition) that is presented to the cognitive agent. Therefore, in order to preserve the intent of exposing what role these “representational” structures or processes in the cognitive agent play, changes to representations must be made “within” the cognitive agent, while at the same time *not* changing any aspect of a domain instance presented to the model or making any changes to the domain definition itself.

One of the important differences between the Class 1 and Class 2 model approaches to representation can now be made explicit. In Class 1 models, the “content” of the representations was suggested on the basis of how we might read the linguistic labels on the components of a representation structure, and interpret them as being “about” what those linguistic labels suggested to us. As shown above, it was found that the only role the labels played in the processes were in comparisons of identity. In Class 2 models, however, there is now a specific domain that representations are intended to be about. This is not to say that labels can’t still mislead us into thinking more is going on than there is — it is therefore still useful to replace or ignore the names given to the representation mechanisms and their components, and instead see what they *do*. But this highlights another important difference between how we go about employing the “representation replacement/change” technique because now labels typically don’t play the role they did in identity matches — instead, the representational units have functional roles in system processing (i.e., the representational units are specific rules for what to do based on what the current state of the system is and what has been “presented” to it).

So now the question is: *How* should we change the representational structures or processes inside the cognitive agent in order to see what affects they have on system processing and on the relation of the representations to the presented situation (based on the domain definition)? Recall that in the HLP models, there is a *representational grammar* which describes the set of rules or constraints that the system follows in the kinds of representation structures it can build, given the current state of the of the Workspace (the structures present in the Workspace represent the sum result of all representation structure building, change, or destruction activity up to the current state). This representational grammar, in turn, is defined as the set of possible functions which can lead to activity in representation construction; these functions are found in the codelets, the Slipnet concept-nodes (and their functional relations), and the functional interrelations between the codelets and the Slipnet nodes. Thus, in order to employ the “representation replacement/change” technique, changes should be made to whatever might affect the representational grammar — and therefore, in the HLP models, changes will be made to either the codelets, the Slipnet, or the relationship between the two.

Now I can employ the representation change/replacement technique to the Class 2 models. Consider a case in which one of Copycat’s representational units is consistently changed or removed — suppose that every instance of Copycat’s ‘a’-recognizer was changed to be a “@”-recognizer. This altered Copycat is then presented with the following “problem situation” (Figure 3.16):

Figure 3.16 - “abc” example

The altered Copycat has no way of ever recognizing that the letter-strings 1 and 2 contain **a**'s. What results is a kind of “representational aphasia”: Copycat has lost the ability to ever recognize **a**'s; and since @'s never appear in the letter-string domain, Copycat's new talent will never be realized. Of course, the same effect could also be rendered by simply removing the 'a'-recognizer codelets.⁷⁶ In these cases the altered Copycat would still function, but would act as if it were presented with this situation (Figure 3.17):

Figure 3.17 - What the altered Copycat ‘sees’ of the “abc” example

Copycat would probably still be able to arrive at a mapping, and even the mapping rule that was most likely prior to the “aphasia” (the rule being: “change rightmost letter of letter-string to it's successor”).

More interesting kinds of aphasias could be induced (some more abstract than others) by removing other codelets or nodes from the Slipnet, or altering the functional connections between the codelets and the Slipnet. For example, removing the 'c' concept-node in the Slipnet would make the discovery of the “correct” mapping rule in the full “**abc** example” (Figure 3.16) impossible because Copycat could never know what the successor to **c** is — that piece of knowledge has been removed from Copycat. Even more drastic damage could be done by removing the codelets that proposed the mapping rules in the first place — in these cases, Copycat would never stop running because no point could be reached in which a mapping could be proposed on the basis of a transformation rule. It is interesting to see the wide range of affects such changes have on Copycat's potential analogy-making behavior. The main point to be made, however, is that representation does not reduce to identity of comparison between linguistic labels, as it was found to reduce to in the Class 1 models. Instead, the representational roles played by the various functions in the codelets and Slipnet are varied and diverse.

Treatments of representational grammars and domain-style modelling

Clearly, the Class 2 modelling addition of adding a domain that the modelled agent's representation capacities are based on does add an interesting way of grounding what the representations are about (what and how they represent). But, does this technique buy the

⁷⁶ While there is an interesting similarity to aphasias in humans, I am not suggesting that this sheds any light on what actually happens in human cases.

kind of approach to representational content required to have an account of how novel representation could emerge or change? In order to answer this question, it's important to first clear up some misunderstandings about the role of domains in modelling.

There is an argument (discussed in Hofstadter 1995, p.189-190) that modeling cognitive agents which work within domains, or *micro*-domains (suggesting the “anemic” semantic value of such domains), makes it so that the cognitive mechanisms (the functional architecture) of the modelled agent cannot be simply transferred and applied to other domains and therefore isn't a very good explanation. The argument goes like this: For SME, the only constraints in use in different domains is that the representations the Structure-Mapping Engine is presented with are in the format of predicate logic structures (as described above in Section 4) — as long as the labels and relations are assumed to correspond to actual entities, relations, or states of affairs in any given domain (or correspond to what knowledge of such a domain would be), the SME algorithm works just the same. For HLP-based models, on the other hand, the whole set of codelet functions, Slipnet nodes and relations, and the functional relationship between the codelets and Slipnet nodes, needs to be re-worked to fit the specific kind of domain. The conclusion (or, at least the one feared) is that SME is a more robust explanation because of its relative “domain independence,” whereas the HLP models are “domain dependent.”

But something is wrong in the way the comparison has been made in the argument. In an important sense, HLP's architecture is just as “domain independent” as SME's if we consider the representational grammar (coded up in the codelet and Slipnet node functions and their interrelations) of the HLP model as what changes from domain to domain, just as SME's representations (labels and relations in predicate calculus structures) are changed from domain to domain. That is, SME's representational format is *also* a representational grammar. The only difference is that FARG has automated the representational grammar in the model, whereas SMT requires a human to handle the “following of the rules” of the grammar. In this sense, HLP has just as much a general analogical mapping algorithm to offer as SMT: namely, the architecture specified in the outline of the parallel terraced scan influencing and influenced by “knowledge” (concepts) of the domain, and governed by the computational temperature feedback mechanism. While more complicated, this is no more dependent on how knowledge is represented than the SME “local-to-global match of knowledge structures and then mapping” algorithm is on its predicate logic knowledge representation format — *its* representational grammar.

Nonetheless, the difference between Class 1 and Class 2 models in the relative “amount” of cognitive architecture affected by the representation of specific knowledge about a domain does still demonstrate an interesting difference between the SMT and HLP approaches to semantics. The difference in approaches is in the extent to which “knowledge” is directly implicated into the mapping processes and the computational treatment of semantics in representations. There are two additions that the Class 2 models make to the handling of semantics in a model's treatment of representation:

(1) Class 2 models add an interesting semantics to handling representation by making a distinction between *different roles* that representations play; namely, the difference between: (a) the role of representations as concepts and (b) the role of representations as structures representing what the current situation is “seen as” (a perceptual role in the sense of demonstrating the agent's “conceptualization” of the current situation). The

interesting addition not found in SME is that either of these “representation kinds” have distinctly different kinds of roles in further system processing (i.e., different dimensions from the perspective of a kind of process semantics — the semantics is determined on the basis of what role in further functioning of the model certain computational processes play). An example is the kinds of codelets that are spawned (the (b)-kind of representations) on the *basis* of concept activation (the (a)-kind of representations). In SME, representations play a role in terms of how they are compared, but they do not play a role in, for example, changing other representations’ structures. Even in MAC/FAC, which does model long-term memory, no process is suggested by which the representations in long-term memory might play a direct role in current alignment and mapping processes; the only role this long-term memory plays is as the “container” of static representation structures that may be selected for the ensuing alignment and mapping processes.

(2) HLP’s approach to representation construction as defined by a representation grammar is also important because this is where HLP addresses some of the potential *internal* dynamics of representation which may play an important role *within* the analogical comparison processes (in the process of structural alignment of mapping). Namely, more of the semantics of the agent’s knowledge of the domain are directly implicated in the details of the processes involved in the structure-building (alignment and mapping) for the analogy process.

Thus, domain modelling does not entail the impossibility of describing a “domain-general” architecture, and it also allows for a computational account of increased representational dynamics. But it is not clear that the HLP approach to the relation between its representational grammar and its domain has left open enough flexibility and control for representation emergence or change. In Copycat, for example, there is a strict relationship between the letter-domain definition and Copycat’s representation capacity: while the domain definition serves to set the terms for the kinds of situations Copycat can be presented with and how Copycat might come to represent those situations, the distinction between what is distinctly Copycat’s domain and what is its representation capacity is very fuzzy. In an important sense, Copycat “knows” all that there is to know of the world that it works in: of what possible transformation rules there could ever be. It is this last point that I believe is one of the key pieces of evidence to unravelling the presence of handcoding within Class 2 models as well.

Handcoding

It is clear that the handcoding of “where analogies come from”⁷⁷ is not just a result of modelling within a micro-domain. I believe that handcoding of this type *can* be avoided

⁷⁷ That is, the involvement of the researcher in the model setup to such an extent that key mechanisms which are involved in actual human cognitive capacities to make analogies are bypassed in the proposed model of such capacities — thus, the analogies come from (at least, to a larger extent than we’d like) the human who set the program up, and not the model.

while still working within such domains.⁷⁸ Rather, the issue of handcoding has to do with the treatment of the *relations between* the representation grammar and the domain(s) represented — specifically, how *content* gets determined.⁷⁹ This is made clear by first investigating handcoding from the Chalmers *et al.* perspective.

Handcoding of the kind Chalmers et al. identify in SME

It is important to note that the criteria employed by Chalmers *et al.* (1992) for identifying handcoding in SME can be employed to identify the same kind of handcoding in HLP models — models that were specifically designed to avoid such handcoding. HLP's design methodology (involving the relation between the cognitive agent architecture and the domain definition, and the subsequent impact of the domain definition on the details of the modelled cognitive agent's architecture) has the following impact on representation and analogy capacity:

(1) The domain determines a whole class of kinds of possible analogies that might be made (based on the ontology — what exists and what the nature is of what exists — *and* the rules for problem presentation). (Copycat, for example, does not make analogies about table-ware arrangements; and Tabletop does not make analogies based on letter-strings.)

(2) The representational grammar, which is based on that domain ontology and the definition of what kinds of presented situations there can be, in turn, defines *all* the possible analogies that *can ever* be made by the HLP cognitive agent modelled (These possibilities exist as the set of all the potential combinations of structures that could lead to the eventual employment of a “translated rule” to finish the alignment and mapping process — the completion of an analogy)

What this amounts to is that once the domain has been chosen, it is entirely up to the creators of the specific HLP model to *hand-code* the kinds of necessary relations between the representational grammar and any possibly presented problem-situations in order for the model to make any analogies — and such hand-coding *determines* what analogies can be made. Thus, in an important sense, the researcher is directly involved in determining what analogies might be made, just as the case in SME-based models with the determination of the structured representations. Certainly, there is more flexibility in the HLP model (primarily due to the stochastic-based search method for the optimal analogy for a given situation) — and this is interesting and important (and therefore, *not* to be down-played, just as the mechanisms SME demonstrates are not to be down-played because of their handcoding) — but it is *not* a complete escape of the very kind of handcoding which Chalmers *et al.* (1992) and Hofstadter & FARG (1995) accuse the SME-based models and methodology of committing.

⁷⁸ Thus, the argument for the *desirability* of micro-domain modelling because of its simplification and clarity of presentation of the mechanisms involved does still hold; but not to the point of also disregarding the importance of modelling in domain-independent fashions as well.

⁷⁹ Note: as I will discuss in the next chapter, this does *not* entail that “hooking up the cognitive agent to the world” in the sense of Harnad's (1990) symbol grounding is the solution.

Handcoding with respect to the emergence or change of the representation grammar

I am, however, interested in handcoding of a slightly different sort — a kind of handcoding which I believe may have been feeding the intuitions of Hofstadter and Chalmers *et al.* (or should have been) and which I believe makes more clear a gap in current explanations of analogy: the lack of an account of the possibility of discovering a *novel* way of representing on the basis of, or in the process of, making an analogy — therefore leading directly or potentially to *novel analogies*: the goal of creative analogy research.⁸⁰

As I have already shown, SME-based models rely on handcoding which denies an account of representation emergence and change. HLP models, however, also depend on handcoding to avoid an account of the mechanisms responsible for this capacity. Most of the groundwork for demonstrating this has already been laid, and requires only the following summary: The representational grammar is made up of representational atoms (in the sense of the atomic rules — functions — and their criteria for application in the process of representation construction); *No* mechanism is offered, however, for how these fundamental representational atoms might be *changed* (there is only flexibility in whether they should be chosen to be possibly applied, not flexibility regarding whether they in fact *will* or *will not* be applied, or applied *in a different way*), or how a fundamentally new kind of representational atom might *emerge* (be *added* to the basic representational grammar). At this point, the *only* way such atoms might be changed or created is *by the researcher*, who sets the relations up between the representational grammar and the domain.

So, having a domain allows for content to be considered via independent criteria. But that in itself does not guarantee the possibility of extensibility of the agent's representing capabilities — which, as I argue, must be a *potential* in all basic analogy-making capability (this potential manifesting in anywhere from minimal learning to profound discovery). Also, this fundamental emergence and change cannot be pushed-off as *solely* a problem of “low-level” perception, which (as described above at the beginning of Section 5.1 in presenting the HLP approach) is where initial processing of “raw sensory information” takes place: emergence and change is also possible within the realm *high-level* perception — that is, not just changes in codelet functions, but *also* changes to concepts themselves (as depicted in the Slipnet).

What emergence or change might be in the HLP approach to representation

It is important to consider what an example of such emergence or change of representation would be in HLP models. After all, if it is a simple matter of modification or addition to the HLP models, then this is not a compelling problem. The obvious task is to get a new Slipnet node, or change Slipnet links (not just changing link “conceptual distance”), or get new codelets (new constructors or new destructors or new kinds of rule proposals, etc.) or change their current functions, or establish a new relationship between influences of codelets on Slipnet and *vice versa*, etc. What could these changes amount

⁸⁰ There is, of course, more to creativity than novelty. Nonetheless, novelty is a crucial ingredient to creativity and is yet to be accounted for.

to in the Copycat or Tabletop domain? The domain could be the same, but new kinds of relations or categories could be attributed to it, and therefore, new kinds of analogical comparisons could be “discovered.” To come up with a specific example of what a discovery might be, it is easier to think of a more primitive version of Copycat: “Primi-Cat” — a -cat that does not have the robust representing capacity that Copycat currently has with respect to the letter domain.⁸¹

For example, suppose Primi-Cat did not have the notion of “opposite,” and therefore lacked the associated Slipnet link-label and Codelet recognizers and destructors. This leaves room for a new and interesting discovery, but then a new question is raised: What is going to motivate Copycat to produce such a new representation, not to mention how? (An account of motivation is probably related to an account of how.) Here we can see that more has to be added than just some more Slipnet connections or codelet types.

(A related question that sheds some more light on the predicament is: How do we judge the significance of what has been discovered? — or, How might the modelled agent judge the significance of what it has discovered? So far, Hofstadter, Mitchell and French have relied on our independent judgements of what are significant (i.e., what we judge as similar and what we find as significant and interesting relations in our world), as coded up *by the human* into the Slipnets and codelets of Copycat (for the Alphabet-domain) or Tabletop (the Tableworld-domain). In real-life, our judgements of significance are largely determined by our experiences in the world of what is successful (what allows us to do new things, or get nice feedback from people, or is the case *by definition*, etc.); these judgements are therefore a determination of complex relationships between our goals and feedback from the environment. Therefore, to not *handcode* what counts as significant will depend on determining what the world is, and then what the agent’s goals might be relative to how it can act on the world, as well as what feedback the world gives and what that feedback “means” to the agent. As I will show in Chapter 4, these will turn out to be significant aspects of basic fundamental representation — aspects not addressable in the current HLP models.)

I’m not yet ready to address any of the above issues (although in Chapter 4 I will present a foundation that suggests the direction to go in to find such answers). Again, my sole purpose here is to establish how such emergence or change of representing capacity is simply possible in the first place. Handcoding of this sort will also necessarily hold for any Class 2 models in which the representational grammar of the cognitive agent is entirely determined by the domain definition and the functional components of the agent’s “representations,” and as long as no mechanism is provided for how the representational grammar itself might be changed or extended.

A final comment on the treatment of handcoding and learning by Chalmers et al.

Finally, it is interesting to note that Chalmers *et al.* are explicit that representation emergence or change is not involved in analogy-making, and defend their handcoding, claiming that, “... it is clear that human beings at any given time have a fixed repertoire of

⁸¹ It is hard to think of believable new categories that Copycat could learn beyond what it already knows about its given letter-domain. In an interesting sense, Copycat knows “too much” compared to what its domain has to offer for any new discovery to seem interesting — at least from our own perspective..

mechanisms available to the perceptual process” (Chalmers *et al.*, 1992, p.209). That is, *fixed perceptual mechanisms* have enough flexibility to capture all the dynamics of representation-building that are involved in analogy-making — and those perceptual mechanisms themselves could never change during analogical processing. Chalmers *et al.* add:

“One might justifiably ask where these [fixed *perceptual mechanisms*], and the corresponding mechanisms in Copycat, come from, but this would be a question about *learning*. Copycat does not improve from one run to the next. It would be a very interesting further step to incorporate learning processes into Copycat, but at present the program should be taken as a model of perceptual processes in an individual agent at a particular time.”
(Chalmers *et al.*, 1992, p.209)

Two things are particularly troubling about this statement. First, “learning” for Chalmers *et al.* seems to be just a “memory” of what paths were taken in certain previous “similar” circumstances — a sort of biasing of Copycat’s behavior towards certain solution paths when the problem is seen as bearing similarity to past successful solutions. But certainly this does *not* capture the idea of learning something fundamentally new with respect to representation construction capacity: such as a new kind of concept or codelet function. If they mean something more than this, they have yet to make it clear. And again, it’s not just a matter of emergence and change in the fine details of low-level perception, as this emergence and change can (and probably mostly does) take place in *concepts* and their structure-building function — that is, *in* high-level perception. (Hofstadter & FARG (1995) do discuss what some additions to the Copycat architecture might be — a *Metacat* — but these don’t address these basic emergence or change issues, but rather, more complex supervision of the already fixed construction capacity.)

The second worry has more to do with the style of argument made by Chalmers *et al.* in their use of the identification of handcoding — use that is damaging to the productivity that actual recognition of handcoding can have. Chalmers *et al.* seem to be arbitrarily drawing the line at where analogy-based phenomena begin and end. Therefore, they argue, it is excusable for Copycat to not account for learning, but inexcusable for SME to not have high-level perception and still be considered to be making analogies (i.e., for SME to not include the role of the construction of representational structure during alignment and mapping). My concern is not that such lines aren’t useful to draw — they clearly are — but that Chalmers *et al.*’s chief argument *depends* on there being such a line. By that same line of reasoning, all of the HLP models likewise fail because of their handcoding of the representational grammar, which amounts to a lack of an account of the deeper kind of representational dynamics involved in analogy-making. Clearly both models have involved handcoding, and clearly there is much more to be learned about analogy. But to claim that SME’s handcoding entails that we haven’t learned something centrally important about analogy — what it is that has been learned, thus showing how successive models are adding to our understanding (such as the notions of progressive alignment and mapping of structure, which HLP also assumes is true, whether they admit it or not).

7 - The Relational Shift in development, Knowledge Change, and Phineas

In this Section I will consider some interesting empirical findings obtained from studies on the development of the capacity to make analogies. I will focus on one particular result and discuss how handcoding in present analogy models preclude a complete explanation of this phenomenon, even though it is widely agreed that the analogical similarity comparison mechanism itself is central to this development. To make this investigation complete, I will consider some additional SME-based models, most notably Phineas, and demonstrate how they are likewise limited by relying on handcoding as described in the previous section, and therefore cannot propose an adequate explanation for the shift. One function of the Phineas architecture, however, suggests an interesting departure from all the other models proposed so far; I will close this section with a discussion of this feature.

7.1 - The relational shift

Researchers interested in the development of the ability to make analogies have found that children go through a *relational shift* that marks a shift from making similarity comparisons on the basis of object features alone (e.g., a round red ball is like a round red apple) to similarity comparisons made on the basis of shared relations (e.g., a dollhouse is like a flowerpot if both are the largest items in two sets of objects). Gentner, Rattermann, Markman and Kotovsky (1995; hereafter, Gentner *et al.*) utilize SMT to explain the relational shift as the result of changes in the child's knowledge of the world — in how the child represents her world. Although their theoretical claims are strongly supported by the data, I argue that SMT's computational explanation of the shift, based on SME simulations, leaves unexplained many crucial details of the mechanisms for the change of representation, the proposed key to the occurrence of the relational shift. Specifically, their account provides us with evidence of *what* kinds of structural features of representation must change for the shift to take place (assuming SMT is correct) and a method for determining *when* these changes occur; but the crucial account of *how* these changes occur is skeletal at best. Furthermore, it seems unlikely that SMT's current computational approach in the SME-based modelling of representation will easily be extendable to account for the requisite changes proposed. I claim that it is the reliance on handcoding (and therefore the lack of an account of representation emergence and change), as explicated above, that precludes SME from full explanation of how the shift occurs — this discussion thus serves as an excellent example of both the power and limits of the SMT approach to modelling representation.

Utilizing SMT, Gentner *et al.* claim that the relational shift occurs as a result of three mechanisms: (1) changes in knowledge, (2) language learning, and (3) the process of similarity comparison itself. More specifically, knowledge change is the general mechanism for the shift, and language learning and repeated similarity comparisons are specific mechanisms that may cause knowledge change. The following are summaries of some of the key supporting results.

(1) *Change in knowledge*: This is the central claim of Gentner *et al.*'s paper. It contrasts with two other competing accounts which alternately propose that the shift is a result of (a) global maturation of cognitive competence (Inhelder & Piaget, 1958; Piaget,

Montangero & Billeter, 1977) and global increase of information processing capacity (Halford, 1992). Gentner *et al.* argue that these latter proposals are not supported by developmental data, according to two lines of argument: (i) children show considerable analogical ability (and thus have made the relational shift) if the domain is familiar (Gentner, 1977a,b; Brown, 1989, 1990; Brown & Kane, 1988); and (ii) the relational shift is demonstrated to occur at different ages (Gentner & Toupin, 1986; Gentner & Rattermann, 1991; Kolstad & Baillargeon, 1991). Gentner *et al.* point out that while this is strong evidence in favor of the knowledge change thesis, more work needs to be done to clearly differentiate and establish the knowledge change thesis; subsequent investigation in the Gentner *et al.* paper provides such justification by demonstrating the next two mechanisms.

(2) *Language Learning*: The effect of language learning to induce relational comparisons was demonstrated by a set of experiments in which children were urged to make comparisons on the basis of similar relations: e.g., the “largest” object in either of two sets. The experiments had two manipulations. The first manipulation, the *literal similarity* task, involved a setup in which the intended object in the target set shared exactly the same features and relations with the example set. In the second manipulation, the *cross-dimensional*⁸² task, feature similarity and relational similarity conflicted. For example, a coffee mug was middle sized in the example set, but the identical coffee mug in the target set was the smallest of its set; hence, despite the identical features of the two objects, their relations (based on relative size) to the other objects in the set differed. Younger children (3-year-olds) initially made only feature-based comparisons between sets of objects, so that they performed well on the literal similarity tasks, but very poorly on the cross-mapping tasks. In another set of experiments, children were urged to use common relational terms to describe the relations between objects in a set (e.g., “Daddy-Mommy-Baby” for “Large-Medium-Small”). Children were already familiar with these relational terms and had no trouble applying them to describe object sizes in the variety of object presentations. In these experiments, performance of 3-year-old participants was dramatically improved when relational terms were explicitly used to label the objects during the experiment (Rattermann & Gentner, 1990; Rattermann, Gentner, & DeLoache, 1990). Clearly, the use of the relational terms helped the young children to make the desired relational comparisons.

(3) *Similarity Comparisons*: According to Gentner *et al.*, the similarity comparison process itself can promote the development of analogy by drawing attention to relational structure, and can likewise help in the acquisition of knowledge of relations which can be used in future comparisons. In this set of experiments (Kotovsky & Gentner, 1990, 1994), children of various ages (4, 6, and 8 years) were shown three pictures, each containing three figures of various sizes and darkness. Children were instructed to choose which of two alternatives was most similar to a designated standard. The “correct” alternative always shared the same higher-order perceptual relation, while the other alternative did not (see Figure 3.18; it was antecedently determined that adults

⁸² This is essentially the same kind of cross-domain situation that I discussed in Section 4.2 and the second half of Section 5.2; the issue of criterial identity still applies here: the SME models still assume the appropriate representational distinctions between features and relations and the weighting of relations (i.e., ignoring features) in comparison

consistently favored the higher-order relational matches). The experimenters manipulated dimension (e.g., *same*: both pictures had “increasing” values in their dimension, versus *cross*:- one picture’s figures changed in size while the other picture’s in darkness), the relations between figures within each picture (e.g., *symmetry*: the sizes varied large-small-large, versus *monotonic increase*: the figures varied from light-medium-dark), and polarity (e.g., *same*: both pictures had “increasing” values in their dimension (both in the same direction: e.g., left-to-right), versus *opposite*: one increased and the others decreased).

Figure 3.18 - Example of trial presentation

Figure 3.19 - Example of Same-dimension vs. Cross-dimension trials

While polarity affected ease of performance, I will only consider the same-polarity condition and focus on the contrast of performance between the trials whose relation

between standard and the correct alternative was same or cross-dimensional (Figure 3.19, above).

Children generally found same-dimension trials easier to match than cross-dimension trials (even 4-year-olds averaged 68% correct in the same-dimension condition). When materials were presented in random combinations of the same or cross-dimension, four-year-olds succeeded in identifying only 49% of the correct cross-dimensional relational matches.

However, when the polarity/dimension trials were *blocked* (I will subsequently refer to this as the *blocked experiment* condition), such that they saw all of the same-dimension trials *before* all of the cross-dimension trials, their performance in cross-dimension trials increased to 60%. This blocking effect was even more dramatic for in a comparison between children who in either experiment (blocked or random presentation) demonstrated proficiency in at least the easier same-dimension conditions. “Proficiency” meant that the child was able to perform at least above 75% total correct choices in the same-dimension condition. When the trials were not blocked (i.e., same and cross-dimension trials were presented randomly), even children who performed above the 75% criterion on the same-dimension trials were correct on only 48% of the cross-dimension trials. In contrast, when the same-dimension trials were blocked initially, children who were at least 75% correct on the same-dimension trials went on to choose correctly on 80% of the cross-dimensional trials (Gentner *et al.*, 1995, p.288)! This suggests that just having consistent practice with the simpler same-dimension trials first allowed the children to acquire some relational-recognition skill that could then be transferred to or set the stage for better performance in the more difficult cases; such advantage was lost when the consistent practice in simpler conditions was eliminated by the random order of difficulty level.

7.2 - The SME simulations of the relational shift

The three proposals for the cause of the shift were further substantiated by simulations carried out using SME. Gentner *et al.* carried out two sets of SME-based simulations to test the claim that the shift is a result of knowledge change. These simulations demonstrated two kinds of knowledge change that could account for the relational shifts observed above: structural augmentation and re-representation.

In the first set, modelling *structural augmentation*, SME was initially run with only representations of first-order relations between objects; this was done to simulate the assumption that young children lack higher-order relational structure in their representations. As predicated by SMT, SME’s behavior was very similar to their findings that three-year-olds typically respond only to object feature similarity. The representations of the first run were then augmented with higher-order relations, which take lower-order relations as arguments, and then input to SME; these new representations were used to model the assumed augmented relational structure available to older children and adults. Again, as predicted, SME’s behavior concurred with the performance of older children and adults, who attended to relations. The comparison between SME’s performance in the relation-poor and relation-rich representation conditions supports the claim that augmentation of relational structure leads to relation-based comparison, just as older children and adults did in contrast to younger children.

Gentner *et al.*, in conjunction with the experiments carried out in support of their second proposal for the shift, claim that it is this kind of structural augmentation that language learning may cause in representation.

The second simulation set modelled *re-representation*. This kind of knowledge change involves a shift from a dimension-specific to a dimension-general representation of relations. For example, the representation of the relation BIGGER[a,b] asserts that ‘a is bigger than b’, but does not allow for comparison with the relation DARKER[x,y] (‘x is darker than y’), even though the two relations implicitly involve a “greater-than” contrast between their constituent components (i.e., ‘b is *greater* in size than a’ and ‘y is *greater* in darkness than x’) — recall: predicates must be identical (bear the same name) for SME to make a match. In order to draw this relational similarity out, the two may be re-represented to highlight the shared “greater-than” relations, while preserving their dimensional information in terms of relative magnitudes along each dimension: GREATER-THAN[SIZE(a), SIZE(b)] and GREATER-THAN[DARKNESS(x), DARKNESS(y)]. The effects of such re-representation was modelled by first presenting SME with the simpler dimension-specific relational representations for comparison, and contrasted the result with a subsequent presentation of the more “analytic” dimension-general representations which share the higher-order relations relating magnitude values within the two different dimensions. Like a four-year-old, SME with only the dimension-specific relations gave a higher evaluation for the same-dimensional comparison than for the cross-dimensional comparison. And when the dimension-general encoding was used, SME preferred the cross-dimensional comparison, as older children and adults were found to prefer. Gentner *et al.* propose that such re-representation therefore accounts for the performance differences between age groups; and this re-representation, they claim, comes about as the result of potential language learning affects (e.g., learning to apply domain-general relational labels) as well as repeated exposure to comparison tasks that draw attention to higher-order relations (as evidenced in the experiments supporting Gentner *et al.*’s third proposal, above).

7.3 - SME’s limits - The *What* and the *When*, but not the *How* of SME

Although I generally agree with Gentner *et al.*’s three proposals for the explanation of the shift and believe they are well-supported by the data, I argue that SME does not directly address the key mechanisms that cause the relational shift. In particular, there are three important observations to make.

Gaps in the explanation of representation development

The first observation is that SME’s simulations give us only “time-slice” views of the proposed processes involved in the development of analogy-making (Figure 3.20). This is a direct result of what SME models: representation alignment and mapping *only*. Because SME’s operation and results depend entirely on the representation structures given as input, the source of the relational shift — of *how* it happens — is located in the change of representation *between* SME simulations. Left unexplained (by SME itself) are the *computational* mechanisms that bring about the change in how representations are constructed and stored.

Figure 3.20 - SME's time-slice view of representation-development

The Striking case of the “blocked-experiment” relational shift provides an excellent case study for investigating these mechanisms — because of the short time-frame in which the shift is observed to occur (the shift was induced within a single experimental run, probably not much longer than 20 minutes), it is likely that a smaller number of ancillary causes (such as exposure to new words, new experiences, or long-term reflection) is involved. This means that the similarity comparison mechanism is most likely the primary mechanism at work in producing the shift. According to SMT, the blocked-experiment demonstrates that young children are *unable* to represent the situation according to dimension-general relations prior to the blocked presentation of the simple, same-dimension relational comparisons. In SME's terms, this means that, when prompted by the comparison task, the child is initially unable to build representations of the compared relations as being the *same* (again, SME can only match two predicates if they are identical; Gentner 1983, 1989; Falkenhainer, Forbus & Gentner, 1989).

SMT's proposed mechanisms for representation change, structural augmentation and re-representation, are viable candidates, but the question remains as to *how* these mechanisms actually work. Here we can see SME's explanatory power and limits made clear. As Figure 3.20 shows, SME is a viable tool for investigating certain features of representation (structure) that may be required in situations of comparison and contrast: for example, for contrasting across age differences, or comparing performance between subjects assumed to have different representation structures. That is, SME provides us with potential insight into *what* structural features are required for alignment and mapping, and in conjunction with the experiments, *when* these features are present. However, SME does not give us a process view of the *mechanisms* for change. What happens between the gaps of SME's simulation time-slices must be handled by other mechanisms. I turn now to the second observation to consider the outlook of other SME-based approaches (systems that produce the kind of structure representations which SME can use — that SME depends on) to account for the blocked-experiment results.

SME-based approaches to representation construction and re-representation

SMT proposes that the mechanism for representation change in the blocked-experiment is re-representation: over the course of the blocked trials, the child comes to re-represent the compared relations in dimension-general terms. Re-representation in the blocked-experiment case has two possible accounts:

- (a) *Pre-existing re-representation capacity*: the child *can* antecedently build dimension-general relations, but doesn't recognize the need to do so until (and only until) repeated presentation of the easier same-dimension relational comparisons; or
- (b) *Emergent re-representation capacity*: the child *cannot* antecedently build dimension-general relations, and over the course of repeated same-dimension comparisons, the ability to build dimension-general representation of the relations *emerges*.

Current SME-based models that propose accounts of how representations are built or re-represented include: Phineas (Falkenhainer, 1988, 1990a, 1990b), MAGI (Ferguson, 1994; Ferguson, Aminoff & Gentner, 1996), and MARS (Forbus, Ferguson & Gentner, 1994). In all cases, "raw data" that is input to these representation-building modules is interpreted according to pre-defined rules for classifying antecedently specified input features as predicate and object kinds (e.g., MAGI's GeoRep module encodes vector line drawings into predicate representations of geometric features; Ferguson, 1994; Ferguson, Aminoff & Gentner, 1996). These models, which constitute larger systems that SME is a module in, clearly extend beyond the Class 1 modelling identified above, and are simple instances of Class 2 (simple because there is not the kind of internal-to-the-system representation-building dynamics as found in the HLP models). But like the HLP models, these models clearly rely on the unchanging representation grammar instantiated in them, and the ultimate source of this grammar is the researcher that built the programs.

Re-representation may be similarly modelled by utilizing pre-defined rules for replacement of component predicate structures.⁸³ Phineas (Falkenhainer, 1988, 1990a,b), for example, uses "general domain knowledge" that specifies rules for how certain representation structures are associated with others — these rules allow for the component predicates or objects in one structure to be replaced by others, under the right conditions (Falkenhainer, 1990b).⁸⁴ One could imagine collections of such background

⁸³ To the best of my knowledge, no existing model does explicitly use pre-defined rules for replacement for re-representation of the kind proposed for the relational shift; as I will show, however, such a model is clearly within the grasp of current SME-based modelling technology.

⁸⁴ In a little more detail: the general domain knowledge specifies how structures of objects, attributes and relations (conditions) imply (are associated with) other objects, attributes and relations (other conditions). If current representations of two situations being compared don't match, and it happens that a component of one representation matches a component of some background knowledge, then that component of the compared representation may be replaced by the implied structure associated in background knowledge, resulting in a newly re-represented structure that might subsequently be alignable.

knowledge for relational predicates. Gentner & Wolff (in press) provide an example that could be implemented by this process (called “predicate decomposition”) for how the metaphor “he had a hot anger” can be understood: ‘hot’ and ‘anger’ could each be re-represented as part of an implied “greater-than” comparison (like the example above at the end of Section 7.2). Utilizing Phineas’s background knowledge scheme, ‘anger’ and ‘hot’ could both have associations to greater-than continuums, where ‘anger’ is greater-than ‘flustered’ in the dimension of annoyance, and ‘hot’ is greater-than ‘cool’ in the dimension of temperature. Once both situations share the greater-than predicate, they can be matched. Such background knowledge for the possibility of predicate decomposition, however, must be in place in terms of *antecedently determined* predicate identities *and* rules for association. No computational account has yet been provided for how such predicate decomposition possibilities or predicates themselves arise in the first place. (Although Phineas does offer an interesting proposal for spawning “analogous” new predicates, which I will briefly discuss in a moment, this does not answer the re-representation problem.)

By this method, SME-based models do suggest a partial explanation of the “pre-existing re-representation capacity” account (option (a)) of the blocked-experiment shift: that is, the child may already have the higher-order relation *and* the association rule(s) that stipulates that under the appropriate conditions the current relational representation can be re-represented dimension-generally. Still lacking, however, is an account of why the children did not represent dimension-generally until *after* the repeated and consistent presentation of the easier same-dimension trials. It can’t be merely a matter of task practice because even doubling the trials did not improve the performance (Gentner *et al.*, 1995, p.288). The computational mechanism still unaccounted for is how the pre-existing rules for re-representation (the background knowledge of associated predicate structures) are primed by this specific order of presentation.

Accounting for the “emergent re-representation capacity” (option (b)), however, is clearly beyond the scope of current SME-based models. We need an account of how the predicate categories for the relations, and the rules for replacement or instantiation of already existing relations, get there in the first place. There are two possible different conditions that could require emergence:

- (1) The higher-order, dimension-general relational predicate already exists, but up till now there has not been an association rule stipulating that the dimension-general predicate can be used to re-represent the dimension-specific situation in a dimension-general form. For example, the dimension-general predicate “greater-than” already exists and is associated with the dimensions of size and darkness, but is not yet associated as also applicable to the continuum of temperature; therefore, the agent cannot re-represent a temperature comparison according to the “greater-than” relation — the association rule is the rule that allows for re-representation (when the higher-order predicate already exists) to be carried out. In this situation, this association rule must emerge for re-representation to then be possible for the dimension of temperature.

- (2) The higher-order, dimension-general relational predicate does not exist at all — this will not only require emergence of the predicate itself, but also the emergence of the association rule(s) as well.

In either case, emergence is required, whether of the association rule for replacement or the predicate itself. Currently, all the rules for re-representation and representation-building are pre-determined, as well as any predicate or object category kinds that are to be instantiated during representation-building. And this cannot be solved by adverting to more rules — the rules themselves still require the antecedent existence of the predicate and object category kinds *before* they can designate their instantiation during representation-building.

SME only accepts representations that are composed of related structures of stable, content-identity bearing, atomic units. In Chapter 4, I will be presenting foundational work on the nature of representation which argues that this kind of representation cannot constitute a foundational form of representation: such units taken as primitives, cannot give rise to fundamentally new kinds of primitives (and therefore, not fundamentally new kinds of relations) — thus, they are either innate primitives or depend on some deeper level of representation yet unspecified (Bickhard, 1993; Bickhard & Terveen, 1995). The requirement of a basic set of fixed and unchanging primitives has even been challenged by recent work in perceptual learning, which suggests that even our low-level perceptual features, once believed to be fixed after a critical period of learning, may change under new categorization pressures (Schyns, Goldstone & Thibaut, in press; Goldstone, Schyns & Medin, 1997; Goldstone, Steyvers, Spencer-Smith & Kersten, in press).

What we need to have for an explanation of the induced relational shift in the blocked-experiment case is an account of the subtending mechanisms which give rise to and maintain stable, content-identity bearing, atomic representations (or predicates and objects) in order to account for emergence of the kind of representations required by SME (or, alternately, if SME is going to be maintained as a general account of the core process of similarity comparison and such representations are rare or non-existent, then this is a grave problem for SME). In other words, we need to know, for example, how the capacity to produce stable, content-identity bearing, atomic representation of a relation like “greater-than” emerges (and gets associated so that future re-representation is possible in other domains).

Finally, I believe there is a strong argument that, although it is entirely possible that re-representation in the blocked-experiment case could involve either (a) pre-existing or (b) emergent re-representation accounts, it is highly unlikely that *only* the pre-existing conditions occurs: (i) First, it is implausible that for *every* case in which re-representation is required, there *already* exists the requisite knowledge of how to re-represent.⁸⁵ (ii) Second, since the ability to re-represent has to be acquired at *some* point, it seems strange to require that it not occur when it is needed most: *while trying to solve the very problems that require such re-representation*. What other conditions would have to prevail, aside from the need for dimension-general relational representations, to provide the information necessary as well as motivate the construction of a specific re-representation

⁸⁵ Again, further support for this strand of argument comes from the *low probability argument* (Dietrich, in press).

capacity? (iii) And finally, it is intuitively appealing that emergent re-representation can occur *during comparison* because it directly implicates analogy-making in the deepest form of learning and creativity: where fundamentally new ways of representing the world (e.g., discovering that hot and anger do share a high-level “greater-than” relationship) are acquired.

Can HLP account for the blocked-experiment relational shift?

As already discussed in some detail above (Sections 5 and 6.4), the HLP architecture proposes some powerful tools for accounting for representation construction. However, as noted above, HLP explicitly eschews the need for learning, and this is reflected in the current inability to account for extension of the representational grammar. Thus, while a representation of the situation may be “constructed” by building a “dimension-general” mapping rule, this rule has to already be possible in the antecedently fixed representational grammar coded in the HLP models’ codelets and Slipnet functional relations — while this allows for more representation dynamics, it does not address the core issue of how such re-representation comes to be possible in the first place, and we are faced with the argument directly above that it is implausible that such possibilities for re-representation are somehow created in situations *outside of* the conditions that call for re-representation capacity emergence. So, once again we are back to the same problem: in no way can Copycat or Tabletop’s representation-building possibilities change over a number of exposures to different input conditions. These models currently have no better access to an explanation of the shift — the representational grammar of the HLP models must be characteristically changed by the experiment just as is the case for SME.

7.4 - Phineas and the skolem predicate/process

As I have alluded to already, Phineas does present an interesting proposal for how the representational grammar of an SME-based system might be extended. Phineas is a simulation of abductive reasoning that employs SME as the core similarity-comparison module (Falkenhainer, 1988, 1990a, b; Forbus *et al.*, in press). Phineas’s operative goal is to find “explanations” for observations of qualitative physical descriptions presented to it. The qualitative physical descriptions (both those that Phineas already “knows” as background knowledge and those presented as new observations) are expressed in the *Qualitative Process Theory* formalism (QP-theory for short; Forbus 1984)⁸⁶ — this formalism is a constrained version of the kind of predicate representations that SME can accept, as described above in Section 4.

Explanations are found based on a search of Phineas’s background knowledge of already “understood” theoretical knowledge (stored QP-theory expressions of past experiences and associated process explanations) analogous to the currently observed

⁸⁶ QP-theory is a formalism for representing and reasoning about physical change. A situation is represented as a structure of objects (each object has a set of continuous quantities, such as temperature and pressure, intended to capture aspects of the state of the object), a set of relationships, and a set of process schemas that account for kinds of changes in object quantities and relationships (these schemas have a format specifying conditions for the process to be possible, what entities, relations and quantities they involve, and how they change under given conditions).

situation.⁸⁷ Similar explanations in background knowledge are employed to explain the current observation. In some cases, new objects must be proposed to “fill in the gaps” of the observed target situation based on additional information in the source explanation. It is this latter capacity that suggests an interesting possible account of representation development: the possibility of extending the “vocabulary” of the representational grammar.

(An important note should be made here: Phineas essentially involves two levels of a representational grammar — the base level of QP-theory, the formalism in which all representations are expressed, and the grammar of what current theories (associations of representation predicate/object structures expressed in QP-theory) Phineas has in its background knowledge. I will treat this latter set of associated structures as a representational grammar because it is in terms of these already known theories that Phineas builds its proposed explanations of new situations. I will address this point explicitly, below, when I analyze Phineas’s treatment of possible representational grammar extension.)

Because of Phineas’s complexity, this predicate creation mechanism is best presented in an example, adapted from Falkenhainer (1990a). Phineas is first given background knowledge of many associated physical processes and principles, including an account of water-flow (like the example I used in Section 4.3.1, but with more detail and associated background information about physics). In this case, however, Phineas specifically *lacks* a theory of “heat flow.” Phineas is then presented with a description of the following physical situation (in QP-theory terms): initially, a hot brick is placed in cold water, and over time the temperature of the brick decreases while the water’s temperature increases until, finally, both temperatures are constant and equal. Because the description of change of the input situation best matches the behavioral description of water-flow, the associated theory explaining water-flow (the source) is retrieved for potential mapping.⁸⁸ During the mapping, SME matches-up as many of the conditions, objects, properties and processes as possible between the two situations. The initial alignment results in a partial mapping because the input situation (the target) does not have a corresponding object for “water” — a *substance* that is taking part in the process of flow (namely, *what* is flowing). Phineas thus proposes a possible object that might play the role of water — a *skolem object*⁸⁹: the unknown object is labeled “skolem-water” and is used to fill in the candidate inferences to build the explanation of the target situation.

This results in a number of candidate inferences, including the proposals that there is: (1) a new “contained-fluid relationship,” in which the temperature of the container (the

⁸⁷ A knowledge-intensive version of SME, called *contextual structure-mapping*, is used for comparison of possible candidate matches in the search; the main differences consist of additional constraints for mapping — including a relaxation of certain identity matches based on a combination of incremental mapping and a kind of re-representation along the lines outlined above (Falkenhainer, 1988, 1990b).

⁸⁸ I am passing over the details of how this search process works. Roughly, the search is carried over antecedently determined “abstraction hierarchies” indexing kinds of behavioral descriptions with associated appropriate theories — additional handcoding issues could be raised, but I won’t pursue them here; see Dietrich (in press).

⁸⁹ The term “skolem object” is derived from the standard logical use of a *skolem constant* to denote the existence of an unknown object and enable removal of an existential quantifier.

brick) is proportional to the amount of the substance it contains — this substance (“skolem-water”) is currently unknown but is analogous to the water in the liquid flow situation (and thus picks up analogous associated properties of water, such a “is containable,” “can flow,” “is a liquid,” etc.); and (2) a new process: when two objects differing in temperature are connected by a physical path, the unknown substance continuously flows from the object of higher temperature to the one of lower temperature, at a rate equal to their difference in temperatures. These are all, of course, expressed in QP-theory terms.

These candidate inferences are then passed to the “transfer stage,” at which time Phineas checks to make sure the current proposals are not inconsistent, and then attempts to replace any skolem objects with additional, already known background information. In this case, Phineas finds no known correspondent to the unknown skolem object corresponding to the water in the water-flow situation.⁹⁰ At this point, a new entity token is created for the missing water correspondent; in the actual simulation, a contradiction arose when the mapping was attempted because a liquid, unknown or otherwise, cannot be contained in a solid (the brick) (again, this is based on background knowledge) — thus, the unknown substance is also proposed to be of a new kind of *phase* (something other than “liquid”; this was automatically labeled “Phase-1”). This new token is subsequently defined as of a new phase type, with a new object instance (which happened to be labeled “sk-water-1”, again following Phineas’s prescribed method for labeling). And finally, to complete the transfer, Phineas proposes a new process (automatically labeled “Process-1”) to account for the flow of this unknown substance from the brick to the water. All of these additional proposals stemmed from the initial new token proposal (and making sure it was consistent with background knowledge).

I have elided a number of details and there are many other features of Phineas that are interesting and deserve attention.⁹¹ However, this is sufficient to uncover how the basic mechanism for creating a new token for the representational grammar works (the grammar in this case being the one based on what is theoretically known in background knowledge, expressed in QP-theory terms, and thus a part of Phineas’s explanation-building arsenal). Clearly the operant question is whether this satisfies my quest to find a non-handcoded method for extending a representational grammar: for new ways the agent can represent its world. There are a couple of points to make here.

First, I should place Phineas in the context of the above analysis of Section 6. This is somewhat tricky because any input of the presented situation is “direct,” with no sort of interpretation handled by Phineas as to what that initial input is — for example, an input description of temperatures of two objects converging is clearly understood to its full extent, unlike the case for Copycat, in which the ‘a’ in an input string ‘abc’ must be additionally interpreted as the *predecessor* of ‘b’ in order for processing with respect to ‘predecessor’ to ensue (and it is possible that this relationship is never “seen” — i.e.,

⁹⁰ Phineas is, however, able to resolve some other skolem problems, such as what allows the unknown substance to flow: Phineas determines, based on background knowledge, that the brick and water share a “common-face,” which can serve as a physical path for substance flow.

⁹¹ Such as the method for initial search through the large database (interestingly similar to and different from MAC/FAC; Forbus *et al.*, in press), the map/analyze cycle of verification and qualitative simulation, and the evaluation and revising of proposed explanations.

built). However, because Phineas does involve a relation to the input in which it must make sense of the input (in the sense of proposing an explanation — building an explanation if one doesn't already exist in background knowledge), it is closer to a Class 2 model designation: the modelled agent must make what it does of the world based on what it “knows.”⁹²

Now, to be a non-handcoded account of representational grammar emergence, I need to know (in an objective sense — i.e., not based on what I may read the labels as referring to in my world) what the content of this generated skolem object is (this content, in turn, determines the subsequent constructions that utilize it). There are several answers that might be given which each raise their own additional issues; I address these in turn:

(1) Given that the knowledge in Phineas has no establishable content beyond the label that it has (as determined in the analysis in Section 6, which is just as applicable here), it is not clear exactly what has been derived by adding a new name, even if it is associated to already existing names. Instead, we are faced with a host of “empty symbols” (with possible new empty symbols) that are associated with one-another. The “empty symbol” problem has been discussed by a number of researchers; notably, Block (1986) and Clancey (1992). In particular, Clancey uses the illuminating metaphor that these associated symbols are (treated from the program's perspective) like arranging, combining, comparing and manipulating the pieces to a puzzle with the “picture” side face down, so that all we can manipulate are the shapes — the content (what the puzzle pieces are about — what they picture) is reduced to only how they fit together. The general point is that this seems to violate our intuition that the symbols, if they are about things in the world or are intended to capture what we know of the world based on having the symbol, should give us something *more* than just *differentiation* from one-another.

(2) A slightly different and more interesting interpretation is to “widen our view” of Phineas's operation beyond single input presentations: Phineas, after all is adding to its grammar *because* it got a kind of input that it couldn't explain. We should thus consider Phineas as actually receiving a steady stream of inputs from the world, supposedly about how the world behaves (ignoring for the moment the rather important issue of how this

⁹² If you press on this, the original class distinction I made begins to break down: one could argue that there is no “modelled world” or domain definition of the sense that Copycat has in its letter-domain, unless one were to consider *all* the allowable QP-theory expressible physical situations, in which case SME likewise is in the world of all allowable predicate calculus expressible situations — a Class 2 situation. However, I believe the distinction still stands by holding the original intent that there is some interesting relationship between the agent and what it might make of certain input which is lacking in the base SME model situation that I considered as Class 1 in Section 6.3.

On the other hand, in an important sense Phineas is still just as much encapsulated in its own world as Copycat is in its “complete” modelling of the letter domain — the letter domain definition essentially only exists in the Copycat *model* to the extent that it is coded in Copycat's representational grammar (the codelets & Slipnet). In a deep sense, there is no epistemic distinction between these agents and their domains in the sense that the world exists independently of the agent and “pushes back” — this is primarily the result of the fact that these agents aren't “pushing” in the first place. And again, however, there is yet another wrinkle here because Phineas, in all its glory (with its proposed map/analyze and revision loop), does “push” internally to a limited extent, and this is an important insight (one that's paid little attention to in subsequent SME-based research) — this is a crucial point that will be raised again in Chapter 4.

gets translated into QP-theory). Now Phineas would be continuously updating its knowledge, adding new unknown predicates to deal with new situations, and slowly building its associated web of background knowledge and experiences. We might consider this input as giving Phineas new “information” about the world and subsequently “adding” or categorizing it in the context of what is already known. However, when considering what role the content of all of these previously existing and subsequent newly proposed objects play in Phineas’s further possible activity and explanation construction, we return once again to simple label name identity with certain associations to other names. Using the label replacement/change technique is revealing: in fact, Falkenhainer even claims, in regards to creating the new process, “Process-1,” based on the unknown skolem substance in a new kind of phase that, “... Additionally, [Phineas] proposes the new Process-1, which might be called a *heat flow process*” (Falkenhainer, 1990a; italics from original). I believe that this is subtly misleading — it suggests that somehow the new process (and proposed substance phase unit) somehow has more content than just the name and association (recall the point made in Section 6.1 that it is insufficient to require that an observer stipulate what the content of a new token is in order for it to have that content). At best, it seems that we are back to point (1): a host of associated “empty symbols” — however, with the additional observation that in some sense these empty symbols serve to *differentiate* kinds of input situations, and Phineas can determine when input differentiations are previously observed or new differentiations, and where they fit amongst already associated and differentiated differentiations.

(3) This leads to a third question: what does Phineas know of its differentiations? The only route available would seem to be to consider the content of QP-theory itself, as it is the “language” in which everything is defined — this is the deeper representational grammar which the representational grammar of Phineas’s knowledge and explanation-building capacity is defined in terms of. However, QP-theory itself, to Phineas, is again stipulated by definition — Phineas in no way controls, modifies or extends the QP-formalism, and it has no way of handling when something is presented in an improper QP-theory format, except to return an error message. The move could be made (and has) that this forces us to accept that there is a *language of thought* (Fodor, 1975, 1987, 1990, 1994): a “language” stipulating how content is expressed in structured computational terms, and whose content itself is ultimately innate, static and inaccessible to modification by current living systems (presumably content emergence and change is handled by evolutionary mechanisms). The language of thought proposal has its own host of problems, which I won’t go into here; in the face of these problems, the most popular defense is that it is “the only game in town.” However as I will discuss in Chapter 4, it is not clear that we are forced to make this assumption — there in fact are alternatives (or at least one).

(3a) The move might be made to consider how QP-theory itself might get extended using Phineas’s object-creation mechanism. Even Falkenhainer admits that QP-theory has its own limits: in a footnote he notes, “Modeling friction and resistance in oscillators is a difficult problem in QP theory” (Falkenhainer, 1990a, p.181, footnote 7). Yet, humans still seem to have the capacity to extend their representation capacity in very fundamental ways — we, after all, have made formalisms for accounting for friction that are not so difficult; it seems that we should have an account for how we create and

change our fundamental ways of representing. And to rehearse a point made above: even if this fundamental creation (emergence) and change only happens once in a while, something is wrong if we can't, in principle, account for the possibility.

The possible extension of QP-theory by the currently proposed object-creation mechanism raises a deep issue. In a sense, QP-theory defines the “metaphysics” of what can be expressed in the world: it defines the language for how, e.g., processes of change of physical situations are to be described. QP-theory is essentially a meta-grammar for representation that is predefined and preset. However, an extension story based on the current new-token creation mechanism is only possible by positing a more fundamental grammar: by the current account, the representational grammar token creation must be *in terms of* QP-theory (i.e., in terms of some other, pre-defined grammar). Thus, if we wanted to propose a mechanism for extending QP-theory, we'd have to advert to an additional new, and more foundational grammar in which to express constructions of QP-theory — this grammar itself now has to be posited by definition and is not extendable. Ultimately, this leads to a regress of requiring further grammars in which to define the content of the present grammar if we wish to extend it. I will address in Chapter 4 whether this regress is a red-herring or indicative of a deeper conceptual problem. (Of course, to bring us back to the above predicament, one could argue that QP-theory is just not the *best* “language of thought,” and that another, more suitable language of thought might be found which is extendable to all the possible subjects that humans currently (and could ever) know — but this language appears to be, in principle, not itself extendable and thus necessarily has to be sufficient to all the tasks to which it might be put.)⁹³

(4) Finally, returning to a more concrete issue, how does the representational grammar token-construction of Phineas fare with the challenge, posed back in Section 6.3, of the “revolves-around” predicate discovery on the basis of observed similarities and differences of situations involving circular motions? (*The abstraction problem*) Certainly, this situation is beyond the scope of Phineas's novel representational unit proposal. Currently, the novel tokens can only be proposed to fill object roles in other predicates that are already known and happen to share many other correspondences with the current situation (e.g., that the same kind of “behavior” is shared by the two situations) — only with these correspondences can the already existing predicates be deemed appropriate as a schematic for the new objects, processes, etc., to be proposed. However, the creation of a *new predicate* on the basis of observed situations that share characteristic similarities and differences may have no antecedent template, as I argue is plausible in the case of an agent observing the situation described in Section 6.3. This requires creating a predicate that has no antecedent analogue but is nonetheless needed to account for the variety of roles played by certain component activities observed across the situations. It is clear that Phineas's novel token proposal does not account for such a situation. And, yet, it seems that humans can do such things: at some point, we learn such new predicates based on our experience of the world — and even if rarely in such spectacular form, precluding that possibility entails a shortcoming of the model.

⁹³ There are some additional issues that might be raised; namely, whether this content issue is just a matter of “hooking up” with the world — in which case is more of an implementational detail. But these issues are best left for discussion in Chapter 4.

Furthermore, to make matters more pointed, since the creation of these predicates crucially depends on the recognition of similarities and differences in order to tease out the new kind of relational roles played by components of a situation, it is clear that this new predicate construction has to be directly implicated in, or directly implicates, analogy-based similarity comparison mechanisms. I do not have a complete solution to this problem, but I argue that a solution begins by first accounting for how novel predicates can be proposed in the absence of explicit predicates that antecedently share the same kind of relational format — something that none of the above models can do.

Certainly Phineas and its representational unit-creation mechanism pose interesting issues with respect to representation development which the previous models have not. The primary theme that the above analysis raises is that the issues of what content is and how new content gets established are nonetheless still central problems, despite a mechanism for spawning new representation tokens — and that these problems are clearly related. While the token-spawning mechanism removes the researcher from having to insert new representations in certain appropriate situations, it is unclear what the new tokens actually represent, and we appear to be stuck with relying on interpreting these tokens as having particular contents (as Falkenhainer did in interpreting the new process as “heat flow”), rather than the token themselves having distinct contents which can be objectively determined (independent of necessary observer attribution).

8 - Summary of the shared representational assumptions and handcoding

I have two reasons for having gone through these families of models in such detail with respect to their treatment of content and the processes that operate on the representations to produce analogy-based behavior. First, as I have shown, none of these models have a satisfactory non-handcoded account of representation content emergence or change. The closest is the proposal of the skolem entity-creation process in Phineas, and that ultimately boils down to the addition of a “token place-marker” associated with other tokens — which is not to say that there is absolutely no interesting semantics here, but that this is not sufficient to ground the proposal in an objective, non-handcoded (or interpreter-based) account.

The second point is related to the first, but is more subtle: I’ve made every attempt to expose what the models *do* provide in terms of the kind of dynamics of representation that we want to have in an account of the role of representation in analogy-making. By this point, it should be clear that quite a lot of computational power with respect to dynamics of representational structure construction and comparison is found in either family of models. And if I were not thorough, it might still seem possible that these models provide an answer to the problem of representational content emergence and change. But, by pushing their interpretations to their limits, it should be clear that they don’t.

As suggested above, I believe that the failure of current models to account for representational content emergence and change is not just a matter of making the current modelling approach to representation more complex in the sense of adding more “knowledge” or rules for association or replacement of knowledge units. Instead, I believe there is something wrong with a common fundamental treatment of

representational *content* shared by either approach that falls in Class 1 and Class 2 models, and their extensions. In the next Chapter, I will address the problems associated with this common approach to content and consider an alternate approach to the fundamental nature of representation that has been proposed to avoid this impasse of not being able to account for content emergence and change. Before doing this, however, I need to summarize and characterize what is common to the two classes of models I have reviewed here.

To review, my methodology has been to be as un-presupposing about the nature of content as possible while still being able to characterize how it is treated in the variety of models I have investigated. In Section 6.1, I started with the idea that content is what the representation is said to offer the agent, since the representation is intended to explain what the agent can be said to “know” or currently make of a represented situation (from its psychological perspective). While investigating each model, I developed the notion of a *representational grammar* to characterize how representation is treated in either approach. The representational grammar is the set of rules which stipulates how representational atoms can be manipulated, combined and related, along with the assumption about the stability of the content of the atoms. Depending on the approach to how this grammar is determined and applied and what role it plays in further system processing, we get an idea of how representational content is handled.

The picture that has emerged is that the common quest of the computational accounts of analogical cognition investigated is to capture the *dynamics* of the role of representation in analogical cognition. However, the shared modelling assumption is that such dynamics can be obtained by considering processes that work *on* representational atoms and their relations (structures of these atoms and relations). The role of the identification of the representational grammar has thus served to highlight the differences between the approaches of the two classes of models as well as what they hold in common:

In the Class 1 models, which centered around the “basic” SME-based approach to representation, the representational grammar is the specification of predicate logic structures which stipulate in what form representations and their structures have to be in order for analogical processes to be able to work. What specific structures might be built, however, is handled entirely by the researcher who provides the representation structures, in the predicate logic format, to the model. As shown above, the specific identity of the content of each atomic representation unit or relation boils down to identity in matches between the labels on units and relations, and how those labels are interrelated in “structures.” Any content beyond this has to be attributed by the researcher (an observer of the system).

Class 2 models likewise have a representational grammar, but the grammar and its rules for application are handled by the computational model itself. This is a great improvement in terms of the dynamics of representation captured, and can be observed in the amount of control the modelled cognitive agent has over the specific application of particular representational functions in building a representation of a presented situation in which the identity of aspects of the presented situation and their relations are determined; furthermore, Phineas also introduced how new representation units might be proposed to fill the analogous roles in new situations. Here there is a much stronger sense of the dynamics of structured representation construction on the basis of content

being handled *autonomously* by the modelled agent. Again, however, an impasse is met in any attempt to account for how that grammar might be extended or changed: content *has* to again be attributed by the researcher or interpreter, even though this time the attribution is in terms of the researcher setting up the grammar in the model so that the model can then dynamically apply its grammar to build “representational structures” of the situation presented to it, or the an interpreter is required to determine exactly what was created with a new representation token.

Thus, in both classes, the common assumption is that representational content is pre-established and remains fixed throughout cognitive activity, whether that content is continuous identity of labeled units and relations or continuous and unchanging rules of representation-structure-building functions (whose final structures are assumed to correspond to features of the domain). And, when a new representation unit is proposed, the method so far still leaves exactly what the content of the new unit is undetermined beyond differentiation from already existing representation.

Whatever the mechanism might be that brings about such emergence or change while making an analogy, it is clear that before one can be proposed, an account must be given of *how* it is even *possible* for representation emergence and change. What is needed is a new approach to content; one in which the modelled agent’s system-dynamics somehow autonomously manages its own content in a way that new content can arise and existing content can be modified. This point about being able to autonomously manage such content for emergence and change is important because it entails avoiding the requirement of handcoding with respect to having a researcher specifically add to or modify existing content. This approach, in turn, requires an objective account of content, so that handcoding with respect to researcher interpretation is not required for there to *be* content.

At the same time, it is also clear that some account of representation with *structure* is still required: the metaphor of structure has been profitably used by the SMT and HLP models in order to account for how features of the world come to be represented in systematic and related ways so that processes involving comparisons of representations of these relations can lead to production of representations which demonstrate how situations have similarity on the basis of relations shared. Whether it is *this* interpretation of the metaphor of structure that can be maintained in a new approach to the fundamental nature of representation is still an open question. Nonetheless, it is not enough for me to just propose the possibility for representation emergence and change: I also need to provide at least a glimpse of how an account of such representation can then support the kinds of behavior we observe that exhibits the attention to and judgments of structured relations in the world — as, for example, when we make analogical comparisons.

“the moving world can only be known by what is in motion”
- Heraclitus (Frag. 43)

Chapter 4

The Foundations for Representation Emergence & Change

1 - Introduction

The task of Chapter 4 is to propose a foundation for representation in which it is possible for structured representational content to emerge and change. Chapter 3 has set the explanatory task by the juxtaposition of two goals:

- (1) We need an account of the emergence and change of structured representations in order to account for the kinds of representation dynamics observed in the development of analogy-making ability (e.g., the relational shift; Gentner *et al.*, 1995; Morrison & Lee, 1998) and as a result of (and possibly *during*) analogical comparisons (Camac & Glucksberg, 1984; Dietrich, in press; Gentner *et al.*, 1995; Gentner & Wolf, in press).

and

- (2) This account cannot depend on handcoding — that is, this account cannot necessarily require:
 - (a) A human that has to put a new label or functional unit inside the model in order for new representation (new content) to be present;or
 - (b) An already representing observer to *interpret* what the representation is about (if the content of the representation — what the representation means — *has to be* attributed by an *already* representing observer, who’s own representing power we have not yet explained, then we still lack an explanation of what content there is or how it gets there).

In Chapter 3, I demonstrated that our current best models of analogical cognition do not account for how representation fundamentally emerges or changes because they rely on handcoding of representational content. I characterized the current approach to

representation shared by both Class 1 and Class 2 models as a *representational grammar* that defines the set of rules by which representational atoms (units proposed to play a representational role) can be manipulated, combined, related and compared. The treatment of representational content by this kind of grammar is fundamentally the same whether the representational content of the atoms consists of stable and continuous (i.e., unchanging) identity of labeled units and relations (stipulated to be about entities, relations, or properties of a domain, from the agent's perspective), or continuous and unchanging rules for representation-structure-building functions (whose units for building structures are composed of such labeled units):

In both cases, the content of each atom or relation (i.e., what each atom or relation is *about*) remains stable and consistent throughout any manipulation, combination, relation or comparison (and presumably any other function within the process of similarity comparison).

While I have shown that current models which rely on this representational grammar style of representation don't have an account of representational content emergence or change because of current dependence on handcoding, I have not yet established that it *isn't possible* to give some such account for a representational grammar *within* the current representational grammar approach. The key question that now drives the search for a foundation of an account of emergence and change for the representational grammar is:

Does the representational grammar approach *necessarily* require handcoding for any possibility of representational content emergence or change?

With this question, an important shift takes place in the use of the handcoding critique and what current findings of handcoding with respect to representational content emergence and change might entail. If handcoding of representational content is *necessarily* required for emergence or change within the current representational grammar approach, then this would mean that within the representational grammar approach *no explanation could ever be given* of how content fundamentally emerges or changes.

Thus, the deeper question to now consider is: *Could* current models account for representation emergence and change if they were just built differently, while still relying on the same underlying representational grammar approach for accounting for the role of representation in cognition? Or, more specifically: Is it possible to provide a non-handcoded account⁹⁴ of *how* the content of the representational grammar's atoms gets established? Answering this question requires looking much closer at the underlying foundations of the nature of representation that may be assumed by the representational grammar — foundations which have not yet been sufficiently explored in what I've presented in Chapter 3 for the Class 1 and 2 modelling approaches. (As I will show

⁹⁴ That is, an account independent of reliance on a human for (a) the set-up or operation of the model or (b) the interpretation of any representational components.

below, we have a couple of options.) In the service of answering this question, I will argue for a set of necessary conditions for the emergence and change of structured representational content.

Layout of the Chapter

In order to answer the above questions and establish a suitable positive foundation, three key issues need to be addressed:

(1) What does *emergence* mean here? Up to this point, I have relied on a generally intuitive notion of emergence. However, there are a variety of notions of emergence, and they each have different criteria for what counts as an instance of emergence and how it is explained. There are two senses I will use, ontological and temporal emergence, and they need to be distinguished in order to make clear what our explanatory task is.

(2) *What* is to emerge or change? Clearly, I am interested in how representation emerges or changes. But, as noted in Section 6 of Chapter 3, I require objective criteria for distinguishing what representational content is before I can consider how it emerges or changes. This requires positing a theory of representation which provides objective criteria for explaining the components of representation (such as representational content) — and, more importantly, clarifying the ontological commitments made by this theory.

(3) *How* does *structured* representation emerge or change? (And, what is meant by *structure*?) Finally, I need to give an outline of the constraints on processes which result in structured representation emergence and change. This requires explaining how the explanatory metaphor of “structure,” crucial to current accounts of analogical cognition, can be conceived in the representational framework I will adopt, and then explaining how such structure can possibly emerge or change. There are a wide variety of different and distinct processes that may result in particular structures, and emergence and change of each may occur under a number of conditions. My answer to this last question will thus be an outline of some of the possible structures that might be constructed within the framework of representation emergence and change that I will adopt — but this outline is by no means exhaustive of the possibilities. However, the outline will serve to capture representation that is at least minimally structured. This outline will also form the basis of the situated representation framework.

Answers to questions (1), (2), and (3), will provide the framework for an approach to the nature of representation that allows for the possibility of structured representation emergence and change. With this framework I will also be able to characterize how the representational grammar’s atoms might be established, maintained and changed — how they can emerge and change. This account of the stability and maintenance of the representation grammar, however, will necessarily rest on top of underlying system processes.

The sections are composed as follows. In Section 2, I will distinguish two uses of the term “emergence,” ontological and temporal, which I will use in the following sections (this answers question (1)). In Section 3, I will discuss representation in general, including the kinds of features required for explaining representation emergence, change and structure. In particular, I will outline three central features of representation: representational content, epistemic contact and the possibility of system-detectable representational error. In Section 4, I introduce Bickhard’s central insight into the nature

of representation: uncovering the choice between two foundational approaches to the nature of representation, based on the relationship between representational content and epistemic contact. In section 4.1, I discuss the first of the two possible approaches to representation — encodingism — and what entailments this would have if the representational grammar were based on this foundation. Key problems are shown to arise from this variant which make it impossible for fundamental emergence and change; thus, this approach cannot satisfactorily explain the kind of role representation is likely to play in analogy-making and its development (as discussed in Chapter 3). This variant is therefore rejected. In Section 4.2, I introduce another approach to the nature of representation — interactivism — and explain how it does allow for emergence and change of representation, as well as account for the central features of cognitive representation outlined in Section 3 (this answers question (2)). I then consider how the representational grammar would be characterized in interactive terms. Section 5 explores the requirements for an account of the emergence and change of representation structure, based on the interactive representation foundation. This includes reconsidering what “structure” is, and how it can be captured in interactivism and still meet the explanatory demands of current models of analogy (this addresses question (3)). I will accomplish this by presenting a thought experiment that demonstrates the situated representation conception of representation as well as provide an example of how an initial kind of minimal representational structure emerges. Finally, in Section 6, I conclude this dissertation project by summarizing the key accomplishments, and the direction we should go in to build a theory of analogy that is capable of structure representation emergence and change.

2 - Emergence

The notion of emergence has received a lot of attention and, unfortunately, a lot of misuse. Some of the problems derive from the fact that there are a number of different ways in which something may be “emergent,” and each kind of emergence merits different criteria for its explanation. Conflating these kinds leads to confusion. The purpose of this section is to make clear how I have been and will continue (with some additional details and distinctions) to use the term.

First, I should discuss *why* the term emergence is used to characterize any phenomena (along with the accompanying proposal to explain such emergence), as these characterizations presuppose ontological commitments concerning the nature of the phenomena and the explanatory project in general. As a basic assumption, I am pursuing emergence within a framework of *naturalism*: by this I mean that all of the events that take place regarding cognition and representation — including emergence of these phenomena — are empirically accessible features of the world, and can be understood by appeal to the laws and theories of the natural sciences. In this case, this commits me to a single fundamental ontology with respect to the phenomena that are to emerge — meaning that all events in this realm, including emergent ones, are explainable in terms of more fundamental entities, properties or processes of the ontology. This is in contrast to a dualism, in which there are two fundamentally distinct kinds of substances, entities, properties, or processes, etc., in the world, and the presence of any property in one realm

is not explainable in terms of properties the other.⁹⁵

In its most general form, then, explaining emergence within a framework of naturalism requires explaining how we go from the *non*-existence of some kind of entity (or state), property, or process, to the existence of that entity (state), property or process within our ontology. (This definition is clearly quite broad, and it may be that most phenomena in the world are emergent in this sense — this isn't a problem for the definition, but a matter of fact about the ubiquity of emergent phenomena in the world; the real task at hand is to explain *how* these phenomena emerge, keeping in mind that it is likely that quite different kinds of explanation may be required for different phenomena.) Emergence, as I have been using the term up to this point, however, implicitly spans two kinds of explanation of emergence that I will now distinguish: an explanation of *ontological* and an explanation of *temporal* emergence. (There are other possible construals, and further distinctions that can be made within each kind.) The explanation of representational content emergence that I have been discussing thus far will require both (although the latter allows a larger latitude of possible satisfactory explanation — i.e., a variety of possible instances of such emergence — which the former does not; see below). Ontological emergence, however, is explanatorily prior to temporal emergence, meaning that accounts of temporal emergence entail an existing account or understanding of ontological emergence.

Ontological emergence concerns how it is possible for some kind of entity, property or process to exist — the necessary conditions for its existence. As an example, consider the phenomena of fire. Fire is a natural kind of event in the world, but specific conditions are required in order for an instance of fire to exist. Namely, there must be the presence of oxygen, a combustible material, and sufficient heat. With these conditions simultaneously met, fire is produced — the phenomena of fire emerges. The explanation

⁹⁵ There are a variety of extensively developed versions of naturalism in cognitive science, each with important differences (see, e.g., Dennett, 1969, Fodor, 1994, and Lewis, 1983). For a strikingly different example, see Chalmers (1996); he proposes a “naturalistic dualism” with respect to consciousness, in which phenomenal consciousness is distinctly different from any physical aspects of the world, but possibly necessarily correlated with physical states (a kind of law-like correlational epiphenomenalism).

Also, a very important and complicated issue pertains to the status of social or socio-normative cognition, which plays an important downward causal role in complex biological systems, such as humans. My adherence to a single fundamental ontology does not preclude the possibility that certain levels of description of nature will have properties that are not type-reducible to particular underlying ontological categories found in single systems. This is not to endorse a kind of property dualism — rather, under certain conditions, certain kinds of properties (e.g., multi-agent normativity and situation conventions) emerge out of underlying organizations of systems (processes) and their interactions (these emergent properties are properties of system organizations and constitute real fundamental emergence; Bickhard, 1998, and Bickhard & Campbell, 1998a, provide more in-depth discussion) Thus, social and socio-normative features of the world supervene on certain states of affairs involving (multiple) agents in interaction (with each other and the world), not in a single, unitary system (one could raise the issue of multiple agents in a single, larger system; but the base point still remains if they are nonetheless distinct agents within this larger system). An important feature of the social and socio-normative world is that descriptive laws do not follow the same kind of predicative criteria that other facets of our ontology do (thus, there are boundary conditions for the applicability of possibility and necessity in this single ontology); nonetheless, these agents (and the environment) are themselves, at lower levels of description, still part of a single ontology of a physical world governed by a hierarchy of interrelated laws and conditions for law-applicability.

of the ontological emergence of the phenomena of fire (put rather simplistically) is thus constituted as the satisfaction of these conditions.

Another example derives from the discussion in Section 3.5 of Chapter 2. The same idea of ontological emergence holds for certain entities found in SVMs: at a certain SVM descriptive level (e.g., the computational description of the word processor SVM I am using to write this text), certain computational properties — e.g., the function for copying a selection of text — exist as a matter of the particular functional organization of the system as a whole. If the underlying organization of my computer were changed, the functional organization of the system that makes copying a selection of text possible may no longer exist even though all computational properties of a lower-level description of the system (e.g., the definition of a variable) may remain intact. On the other hand, once the system is appropriately organized (i.e., certain necessary conditions are met), the function does then exist (it is available for invocation by the word processor SVM). Although copying selections of text in a word processor do not constitute natural kinds, their status as emergent (as functional organizations which only exist under appropriate conditions; and once those conditions are met, that functional organization does exist) is still the same — and still requires the same kind of explanation involving certain requisite necessary conditions.

There are two additional issues which ontological emergence raises. The first regards the kind of ontology required to explain emergence. Specifically, the difference between substance and process ontologies. For an understanding of emergence in most cases of natural science, a shift has been required from a substance ontology to a process ontology of the phenomena involved (Bickhard & Terveen, 1995). For example, historically there has been a shift from the conception of life as a vital fluid (a substance) to life as a form of open system (a process model). Within a substance ontology, the fundamental substances themselves cannot emerge. For example, the Greeks' four fundamental substances of earth, air, fire, and water could not themselves emerge, but had to be in existence from the beginning. In substance ontologies, the basic substances serve as primitives which cannot themselves be derived from within the ontology itself. In the Greeks' conception of the world, then, fire did not emerge; it was always present in the world (even if in varying "amounts" (concentrations) at different times and in different locations).

Process models, on the other hand, allow for an account of emergence within the ontology itself. The key to understanding the possibility of ontological emergence in process models is to recognize that certain causal powers of systems exist as a manifestation of particular system process organization — as a product of the particular functional organization of a system. Changing the functional organization of system processes will subsequently change the system's causal powers. In this way, novel causal powers of a system may emerge as a result of new (or change of existing) system process organization (Bickhard, 1998; Bickhard & Campbell, 1998a).

How such emergence occurs, however, still requires explanation — such an account is not given for free, arising simply by adopting a process model. The emergence of life, for example, *can* be explained in a process model such as considering an open system dynamics; but *how* this happens — how such system dynamics is possible and maintained (an explanation of ontological emergence), or additionally, how it historically came about (a particular account of temporal emergence, see below) — is still a further

question. This is important because, in an important sense, the “explanation” I gave above of the necessary conditions for the existence of fire (as the presence of sufficient friction (heat), a combustible material and oxygen) is still far from complete — we still need an account of what it is about the satisfaction of these conditions that leads to fire: what particular processes are involved and how they work (e.g., what role does oxygen play in the combustion of a material in a fire). (The subject of Section 4.2, below, will be an account of representation in a process model.)

Some examples of systems which are best described in terms of ontologies of process include: closed system stable processes, such as atoms and molecules, and open system stable processes, such as fire (and life... and representation). Bickhard & Terveen (1995, p.167) point out that these ontologies *do* involve instantiation in material terms, but not in terms of forms of material *types*. For example, a flame is not just the molecules and atoms constituting it — the same material engaged in different interactions would not be a flame, and differing material substrates are in fact involved at each moment in the flame’s existence. The entailments of reliance on a process or substance ontology will factor centrally in the discussion in the following sections.

The second issue regards the dependency of kinds of entities, properties and processes on other entities, properties or processes for their existence. Because emergence depends on pre-existing conditions, including the presence of other entities, properties or processes, a hierarchy of system organization types comes into play, with certain system types depending on others. So, for example, fire depends on the existence of oxygen, some combustible material, and sufficient heat (all interacting, involved in some particular kind of process). These entities, properties or processes may themselves, in turn, rely on other conditions for their existence — for example, oxygen requires specific kinds of atoms to be present within appropriate boundary conditions in order to be bound appropriately.

I will not be addressing issues of what may be the “fundamental ground” of the ontology (e.g., are atoms fundamental substances and everything else is built out of combinations, relations or processes involving atoms? — See Bickhard, 1998, and Bickhard & Campbell, 1998a, for discussion). Instead, I only need to advert to an acceptable prior set of properties and conditions (a certain ontological level) in which the desired emergent entity, property or process does not exist, and then account for how that entity, property or process emerges out of these prior existing entities, properties or processes under certain conditions. Again, using the example of fire: combustible material, oxygen and heat do not themselves in any way “contain” fire⁹⁶, but when they come together under the right conditions, they produce a state or process of fire. This latter point is very important to the notion of explaining ontological emergence, because it is often the case (particularly with phenomena less understood than fire, like representation or intentionality) that certain key feature(s) of the emergent phenomena are assumed to be present in the very components that are relied upon to explain such emergence. This leaves unanswered how these features, which we were attempting to explain the emergence of, got there in the first place (in terms of explanation of certain

⁹⁶ That is, none of these contains in themselves an instance or instantiation of fire — nor do they contain *all* of the necessary causal powers for fire, although each may contain in some sense one or more causal powers necessary for fire.

features, relying on the prior existence of those features begs the question).

Temporal emergence — also referred to as historical (Hendriks-Jansen, 1996) or genetic (Nagel, 1961) explanation — involves an explanation of how the specified conditions for ontological emergence might come to be satisfactorily met. Being an issue of timing (or at least sequence) of events, temporal emergence typically involves an explanation in terms of process. This subspecies of the general concept of emergence, as I defined it above, may be formally described as follows: at time t_1 , some particular instance of an ontological kind does not exist, then at some later time t_n , that kind does exist (even if only temporarily). An explanation of temporal emergence is of what happened after time t_1 to produce the emergent at time t_n .

Typically there are multiple kinds of temporal events that may lead to the emergence of the phenomena. For example, fire may be produced by striking a match, rubbing two sticks together, holding a magnifying glass to dry leaves, igniting a gas stove, etc. Each of these events involves its own sequence, and within each kind of sequence there may be wide variation in constraints on timing, order, and even the particular components involved. Constraints on these variations depends on the emergent phenomena in question and on the particular constraints themselves; for example, for fire, low friction may require a very combustible substance to produce fire, while extremely high friction may require a less combustible substance. Nonetheless, in each case, the conditions for ontological emergence must be satisfied. In this sense, temporal emergence regards the ways in which sufficiency for the existence of the kind in question (i.e., satisfying all of the necessary conditions of ontological emergence) may be obtained.⁹⁷

Why, then, is temporal emergence interesting, and why do I need to account for it? Because only certain kinds of temporal emergence (even if a wide variety within the constraints of ontological emergence) are possible — and the variation of these instances becomes particularly constrained in the case of naturally occurring phenomena, such as temporal emergence of representation in analogical cognition in humans. The two theories of analogy reviewed in Chapter 3 are good examples of attempts to explain the temporal emergence of analogical cognition as it occurs in humans. In an important sense, the SMT and HLP models both produce analogies (if we ignore for the moment the status of the “representations” in the models); but while there may be a number of processes involved in the natural production of analogies in humans, it is likely that only one of the proposals is closer to these natural cases (or possibly neither are close; or they only capture a portion of central requirements for analogy).

More fundamentally, however, is the point that while I may have an explanation of the ontological emergence of representation — the necessary conditions for representation to exist — there still remains the question of what processes or conditions might lead to *new* representation coming into existence (being constructed). For example, suppose we lived in a world in which certain systems have representations, but they never changed, never went away, and new ones never came into existence. In this world, an account still exists of the ontological emergence of these representations (an

⁹⁷ I should note that this is a distinct simplification compared to the more complex issues involved in, for example, evolution and the determination of traits (which are emergent). See Hendriks-Jansen (1996), Millikan (1984, 1993) and Nagel (1961) for discussions of varieties of approaches to establishing natural kinds in historical emergence.

explanation of how, at certain levels of description, representations exist). An additional account, however, is required of the processes involved that might allow for such construction. This temporal notion of emergence has also been implicit in my use of the “emergence and change” terminology up to this point. And it is this kind of emergence that I argue is required for certain kinds of analogy-making to be possible (e.g., that observed in the induced relational shift in the blocked-experimental condition, and in creative analogy-making). (This also drives home that the temporal emergence of certain phenomena may depend on or involve the prior or concurrent emergence of other phenomena.) Below I will present a general framework for, and an example of, such constructive emergence that is commensurate with my account of the ontological emergence of representation. Much more work is required, however, to tease out the specific kinds of temporal emergence of representation that lead to the production of analogies.

Now that I have made the necessary distinctions between the two kinds of emergence that I will need to explain representation emergence as required in certain kinds of analogical cognition, I now turn to discuss general features of representation and present the criteria that a theory of representation should meet.

3 - What representation is for — features of cognitive representation

There are three core features of representation that I will make use of in the following sections. We can extract these features by considering the question: What is representation for? In order for systems to make good (adaptive or successful... or any) decisions about what to do based on properties of the world that are not proximately causal, a system must have some sort of access to those properties. This access is in the form of internal states or processes — representations — which provide the system information about those distal properties. Representation thus serves to mediate our interaction with and cognition about the world. The central property of representation is this “aboutness,” and explaining it requires use of the following two concepts: representational content and epistemic contact.

(1) *Representational content* is what the representation informs the system about what it represents — what information⁹⁸ the representation provides the system about what the representation represents. Importantly, representational content serves as the foundation for characterizing the contribution representationality makes towards explaining what the system *knows* — specifically, what the possessor of the representation knows about what the representation is supposed to represent. Clearly, representational content is a central feature for explaining the aboutness of representation.

(2) *Epistemic contact* is whatever is supposed to make the functioning of a representation appropriate to the current situation — it is what gives a representation contact with or relevant to (an aspect of) the environment. The term “epistemic” is used here because this contact does involve whatever connection there is with some state of affairs external to the representation itself — contact with something that *could* be the

⁹⁸ ... in the general sense of “informing,” not necessarily in the information-theoretic sense of Dretske (1981).

object of knowledge or knowing. However, strictly speaking, such contact does not necessitate specific knowledge or knowing for the system — so does not, by itself, make the system that has such contact have knowledge. Rather, such contact provides the *possibility* of knowing something about the external state of affairs (depending on the nature of the contact). In lieu of more bulky phraseology, such as “contact conferring the possibility of a kind of knowledge of the world,” I use the simpler term “epistemic contact.”

Representational content (1) and epistemic contact (2) can be viewed as two sides of the representational coin — two components of explaining representational *aboutness*: content captures the notion of a representation being about something in terms of *what* the system that has the representation (or is representing) *knows* about what is represented, and contact is *how* the representation is about that which is represented. These two features and how they are related are necessary if we are to explain representation. Without content, representations would not inform the system about anything in the world, regardless of any connections the representation may in fact have with states of affairs in the world. Conversely, without contact, whatever representations did inform the system would have nothing to do about the world — a kind of idealism or solipsism. However, once these properties are together, they successfully complete the full characterization of a representation’s aboutness.

(3) A final feature of representation concerns a representation’s potential failure to get its job done — a failure of successful representation, or a failure of successful informing about what is the case in the world. Thus, an additional property to explain is how it is *possible* to *misrepresent*. However, it is not enough to just account for how a representation can be wrong (e.g., positing “noise” in the signals picked up by our sensory systems). We are also able to *recognize*, at least some times, that we have failed to represent correctly or accurately. Thus, we also need to have an account of how it is possible for the representation-bearing system itself to detect that its representation is in error: *system-detectable representational error*. This latter feature is quite tricky because we can’t simply hold up our internal representation of the world and compare it with what is actually out in the world that it is supposed to represent. To do so is an impossibility, and to propose such a possibility is question-begging precisely because it is our internal representations that explain how we have access to the world (if there is one) in the first place. Historically, the recognition of this dilemma has resulted in a wide variety of metaphysical stances that have taken this as a *reductio* against any possible verification of inner content with how the world is in fact.⁹⁹ This is in fact a dilemma, however, only if there were no other way to obtain the information required to do such checking. Below I demonstrate that there is such an alternative. This capacity for system-detectable representational error is necessary for a number of very important representation-dependent cognitive processes, the most important of which are learning and development.

⁹⁹ Amongst these frameworks are *idealism* (all that exists is our experience — there is nothing else to check our experience against) and *nativism in-principle* (we cannot learn anything fundamentally new — we are born with all the content we can ever have, and experience, at most, teaches us how to arrange and combine this pre-given content to build our conception of the world; this fundamental content itself, however, is not open to change and new fundamental content cannot emerge).

4 - Two approaches to the nature of representation

Mark H. Bickhard (1980b, 1993; Bickhard & Richie, 1983; Bickhard & Terveen, 1995; Morrison, 1997) has highlighted a deep and centrally important issue concerning the fundamental nature of representation: that representational content and epistemic contact are two distinct aspects of representation, and a theory of representation must explain the relation between them. As noted in the previous section, representational content and epistemic contact are two sides of the representational coin: content of a representation explains what a system knows in virtue of having a representation, and contact explains how a representation is somehow about (corresponds to, relates to, etc.) some external state of affairs (external to the representation itself).

The fundamental nature of representation depends on the relation these two aspects of representation have to one another. There are subsequently two fundamental options for explaining the relationship of these two features of representation: (1) they may be taken to be provided by or constituted in *one and the same* entity, property, or functional organization of a system; or (2) they may be taken to be *separate* entities, properties, or functional aspects of a system, and it must then be explained how they are related to each other (more specifically, how the system is functionally organized to support this relation). These two options constitute fundamental approaches to the nature of representation.

Bickhard has labeled option (1) *encodingism*; he and his collaborators have developed an extensive critique of this approach, arguing that it is based on a foundational incoherency, and thus bars explanatory access to key cognitive phenomena (e.g., representation emergence and change — learning and development). Bickhard and his colleagues have also developed an alternative approach to the nature of representation which stems from option (2); this alternate framework is called *interactivism*. In the next two sections I will consider these two approaches to representation and what they entail for an explanation of representation emergence and change within the representational grammar approach; Section 4.1 will discuss encodingism and Section 4.2 will introduce interactivism

4.1 - Encodingism - the fundamental assumption and its entailments¹⁰⁰

The assumption of encodingism

The first approach to the nature of cognitive representation that I will consider is encodingism (the encodingist framework). The deepest assumption that the encodingist framework makes is that epistemic contact *provides* or *delivers* representational

¹⁰⁰ Some of the material in Sections 4.1 and 4.2 appeared previously in Morrison (1997).

content.¹⁰¹ That is, according to encodingism the system knows (has content about) what the representation represents in virtue of the establishment of the epistemic connection (contact) between that representation and something else. This assumption entails that the basic representational unit is an *encoding*. An encoding is a “stand-in” for whatever it is supposed to represent. That is, it is a “stand-in” precisely because the above assumption posits that epistemic contact provides representational content: a representation tells the system what it is a representation of so the representation can be used in place of the actual thing it is standing-in for.

An example of a representational scheme which uses this notion of an encoding as a model of representation is that of a *transducer semantics*. Transduction is technically the transformation of forms of energy, but here I am considering transduction as playing a representational role, following the assumption of encodingism. The basic idea is that the system transducers (such as sense receptors) receive energy from the environment that is in causal correspondence with things of importance in that environment. The transducers then “transduce” (transform) that energy into internal processes or symbols which are then said to represent what externally caused them. In this way, an internal symbol **Y** “stands-in” for an external event **X** if symbol **Y** is created or activated by the transduction of a signal from **X**. Here, the stand-in changes the form and medium of representation, which in turn changes the ways the representation may be manipulated. For example, event **X** might be the presence of light (photon energy), but its transduced stand-in, symbol **Y**, may be electrical or chemical activity in a neural structure, or an atomic unit that could be stored and participate in computational operations (e.g., as a variable value in a computer).

This encoding scheme (and encoding schemes in general) can be generalized to chains of encoding relationships, so that a symbol **Y** can stand-in for the symbol **X**, and a symbol **Z** can stand-in for **Y** — and this relationship is transitive so that in such a chain, **Z** could stand-in for **X**. The crucial aspect of encodings is that these stand-ins represent whatever they represent (carry whatever representational content that they carry), “... by virtue of having borrowed it from whatever they are standing in for” (Bickhard, 1993, p.287). Thus, in the transducer-semantics example, encoded symbol **Y** “borrows” its representational content (what it is about) from the external event **X** — and this “borrowing” (if even possible) comes about from the transduction of the signal from the external event to the activation of the internal symbol.

To reiterate, the encodingist framework assumes that these encodings take epistemic contact to provide or constitute representational content.¹⁰² Thus, an encoding represents a particular thing (carries content of that thing — is about that thing) and somehow

¹⁰¹ This seems to be the only option when considering one of either content or contact providing the other: Having content provide contact seems immediately incoherent, something akin to “having a thought makes a world.” Versions of idealism in fact talk this way (e.g., Berkeley’s view of everything only existing in God’s thoughts), but I will not pursue this further — I am starting from the premise that there is some viewer-independent world that can be known (represented).

¹⁰² As I will discuss below, there are other ways of accounting for encodings without the encodingism assumption of stand-ins (with contact constituting or providing content) as the basis for representation; but this, then, won’t be encodingism, but instead an encoding that is derived from, or dependant upon, some more fundamental form of representation.

informs the system of what that encoding is supposed to represent. In its naked form, it is *this* assumption which probably draws the greatest attention for being problematic — and it is from here that all the problems stem. The key problems with encodingism turn on this group of related questions regarding representational content: what is the nature of this content, where does it originally come from, and how is it carried?

The problem with encodingism

To assume a position that is explicitly or implicitly a part of encodingism is to do at least one of the following:

- (i) To assume that encodings are the essence of representation.
- (ii) To assume that encodings are a logically independent form of representation (that is, to assume that encodings do not rely on any other form of representation to be a representation).

Taking an encodingist position, by accepting one or both of points (i) and (ii), leaves us with the following choice. Either we:

- (a) assume that representation is rendered in terms of encodings with representational content, but give no model of *where* this content comes from originally and *how* these elements can carry it; or
- (b) attempt to actually explain where representational content comes from and how encodings carry it from within encodingism.

Choice (a) is clearly unacceptable: it is precisely representational content that we need to account for — where it comes from and how it's carried. We can't simply assume that encodings have content as it is precisely the story of how that content gets there and is maintained that we need to tell. Without it, we couldn't account for how content could emerge or existing content could change — as I argue is required to explain the kinds of analogy phenomena I discussed in Chapter 3. Furthermore, even if it was assumed that the content of encodings didn't emerge or change in analogy-making, there would still remain the task for the cognitive science and AI programmes, which aspire to explain mental phenomena (the base form of which is representation), of explaining how whatever foundational encodings there are got their content — what the source of that content is, and how those encodings obtained it. Choice (a) in this latter case then leads to a vicious circularity: simply accepting (a) with no model of the source of content for encodings simply assumes what it is that cognitive science and AI are trying to explain — that encodings *have* content; but it is precisely this content and “how to have it” that cognitive science and AI are trying to explain.

Choice (b), while being the only viable alternative, is ultimately doomed to failure because of encodingism's fundamental incoherency: attempts to explain how encodings carry or originally produce representational content *cannot* be done *within* encodingism. Encodings cannot be the fundamental nature of representation because it is logically

impossible to explain by use of encodings alone how it is that they have representational content. Encodings, as construed above, *carry* representational contents, and already established encodings can *provide* representational contents for the formation of some other encoding, but there is no way within encodingism itself for those representational contents to ever arise or themselves change in the first place. There is no account possible of the *emergence* or *change* of representation.

This point deserves more attention. We already know that encodings can transitively pass content along: **Y** can stand in for **X**, and **Z** can in turn stand in for **Y**, so that **Z** can then stand in for **X**. In this chain, **X** is the encoding that is providing content for the other two encodings. At some point, however, this chain of providing content has to stop if we are ever going to capture the *original* provider of content — where it is that representational content comes from. I will call this the *bottom-level* foundation of logically independent representations. At this bottom-level, representations cannot be standing-in for others, and therefore these representations cannot be carrying contents provided by any other representation (to do so would mean that we are not at the bottom level). So this cannot be the answer. The only other recourse, however, is to consider this bottom-level encoding as providing its own content. This amounts to asserting “**X** represents whatever it is that **X** represents” or “**X** stands in for **X**,” both of which are vacuous statements — these statements do not in any way succeed in explaining how **X** is provided with content, and therefore how **X** could be an encoding. Thus, the assumption that encodings can be original providers of content is false, and we are left with no account of where content comes from or what its nature is. Because encodingism assumes an impossibility — that encodings can be content providers — it is incoherent.

The version of the incoherence argument I have just presented was from the perspective of attempting to establish a *provider* of representational content from within encodingism. There is, however, another approach to attempting to make encodings the base of, or a logically independent form of, representation: to consider encodings as interpreted. In this case, there must be an *interpreter* which is interpreting an encoding as being about something — essentially, it is the interpreter that is providing the content (determining what the representation is about). To explain representation from this perspective (particularly, how it emerges and changes), however, we now need to account for the interpreter that is providing an encoding with its content — to account for the origins of content (how it emerges), and how it is applied to encodings. But encodingism requires that we can only account for this interpreter fundamentally in terms of encodings (stand-ins), which in turn need to be interpreted. Thus, we run into a vicious regress of interpreters. This form of the incoherence argument is adopted from J. J. Gibson’s critique of encoding forms of perception — the “homunculus regress problem” (Gibson 1950, 1966, 1979; Bickhard & Richie, 1983, discuss this version in detail). Again, this time from the interpreter perspective, encodingism is based on an incoherent foundation.

The results of encodingism’s incoherence

The result of the incoherence of encodingism forces an important conclusion: since encodingism is incoherent (or programmatically circular, as per the programmatic choice (a)), that entails that the assumptions of encodingism must be wrong: if encodings exist, they cannot be the fundamental nature of representation *and* they cannot be a logically

independent form of representation (this is a denial of both assumptions (i) and (ii), above). This in turn entails that whatever the fundamental nature of representation is, it cannot be the case that representational content is the same as or delivered in virtue of epistemic contact. Representational content must ultimately emerge in some form other than encodings; once this content has emerged, it could *then* be provided for the constitution of *derivative* encodings. Furthermore, irrespective of whether encodingism is a straw man (whether anyone *in fact* holds this position; see Bickhard & Terveen, 1995), the conclusion that encodingism is incoherent and thus that, at root, representational content is not the same as or delivered in virtue of epistemic contact, still holds.¹⁰³

Encodingism, handcoding and the representational grammar

The result of encodingism's incoherence entails that encodings can only account for the transfer¹⁰⁴ of representational content from *already* existing representations with content — that is, while working only with encodings, there is no way to get fundamentally new encodings with new content or alter already existing content. What, then, is the status of the representational grammar? The representational grammar of the two families of models considered above is chiefly characterized as a rule-governed set of units which bear — either by antecedently fixed and unchangeable associative rules set by the model builder or as suggested by some linguistic label — stable (non-changing) representational content. (I.e., each atom or relation of the grammar is stipulated by association or label to be about some object, feature or relation in the world, and this aboutness is assumed to remain the same over manipulations, combinations, relations (to one-another) and comparisons.)¹⁰⁵ This characterization consistently fits with the central “stand-in” definition of an encoding: the representational atoms of the grammar are stand-ins for the objects, features or relations they are intended to represent. Thus, based on the inherent limits of encodings that the above incoherency argument uncovers, new content cannot be obtained and existing content cannot be changed by using the representational grammar itself. The source of any content in the grammar must derive from some other form of representation that is, itself, capable of providing representational content.

¹⁰³ There is another line of argument against encodings that is related to the incoherency argument: The argument that encodingism entails the impossibility of learning. Bickhard & Campbell (1996) discuss encodingism's problems from this perspective in more detail.

¹⁰⁴ ...through some appropriate epistemic contact, such as a definitional or causal correspondence.

¹⁰⁵ The one slightly more complicated exception is the skolem predicate/process that is spawned by Phineas. But, as noted above, this amounts to spawning a “token place-marker” with no clear objective method for determining content; while the (already representing) programmer has been removed from having to insert a new “representational marker” when one is required, we are left with still having to rely on (already representing) observer interpretation of these tokens, rather than the token itself having distinct contents which can be objectively determined. Even positing that the token gets its contents in virtue of some causal connection (e.g., transduction) with the world is not sufficient, as the above regress of encoding content provision makes clear.

Given the above discussion of the status of the representational grammar, I can now uncover the relation between handcoding, encodings and encodingism. From the perspective of computational modelling of representation-based phenomena in general, consider the following conditional: If *all* of the representations in a given model are encodings, then *any content in the model (and how it got there) would have to be provided by the interpreter of the system*. This interpretation is necessarily required because, even though content-attributing associations might be made between encodings (either by the programmer literally coding them in the model as associative rules, or by the program's own capacity to build associative links between encodings — the associations in either case being intended to instantiate a content-transferring epistemic contact), no account of how representational content is originally derived is provided by these associations. At most, there is only the creation of epistemic contact for possible representational content transfer, *were such content to exist*. Even for the Class 2 models, which do in some sense posit a “world” that the agent “makes contact with,” the associations (the epistemic contact) between internal system encodings and the external world cannot count as true transfer of content so that the internal encodings have legitimate content *unless* an account of how content arrives from (i.e., is derived from) the world is given, and this is precisely what is lacking in current models (at best, it is simply assumed to be transferred). In this case, the very same example of the transducer semantics I used above — of using transduction of from-the-world signals as the transfer of content about the world to the internal encoding at the other end of the transduction — serves to make clear that the Class 2 models have offered no more of a solution.¹⁰⁶ Representational content must be derived in some other manner. The only other option is to consider the content as being provided by already representing observers/interpreters of the system.

An observer of the system as the only possible source of representational content (an *observer semantics*¹⁰⁷), however, is no more of a solution. The interpreter's own capacity to attribute representational content or to successfully provide the required epistemic contact with a content provider has not yet been explained — and, according to the incoherency argument, it cannot be explained while relying only on encodings as the source of content. This, in turn, means that:

Reliance on encodings as the basis of representation in a model (i.e., assuming encodingism) entails a *necessary reliance on handcoding* of the representations in that model: if the model itself does not provide content, then the creator or interpreter of the model must provide that content.

¹⁰⁶ And this is the case irrespective of whether the world is: (1) a set of associated labeled units treated as “input,” as is in fact the case with all of the Class 2 models I have reviewed above (e.g., the presented letter-string ‘abc’ or the “qualitative physical signal” that is input as a set of numerical values plus labeled “descriptive” units), or (2) is the actual physical world that is interfaced with by a transducer that takes real physical signals from the environment (e.g., a video camera transducing light into a set of internal-to-the-computer encodings).

¹⁰⁷ See Bickhard (1993, pp.289-291) and Bickhard & Terveen (1995, pp.27-30) for elaborated discussions.

Since the phenomenon we are trying to explain requires an account of representational content emergence or change, as I argue is the case for analogy, this is devastating.

I have already demonstrated in Chapter 3 and above that the current models do, in fact, rely on encodings for the representational grammar with no alternative account of the source of representational content within the model. Thus, the models rely on handcoding of representational content for the representational grammar — and this currently precludes an account of representational content emergence and change. Given this connection between reliance on encodingism entailing necessary reliance on handcoding for representation in a model to have any content, I can now conclude with an answer to the key question posed in Section 1 of this Chapter:

Does the representational grammar approach *necessarily* require handcoding for any possibility of representational content emergence or change?

Answer: If we assume encodingism, then *yes*.

Encodingism, however, is not our only option. In the next section (4.2) I will present the core framework of interactivism and will consider how representational content emergence and change is possible, and where the representational grammar might fit (if at all) in such an account.

Before presenting interactivism, however, three final comments should be made regarding encodingism, handcoding, and explanation. First, in terms of the discussion in Chapter 2 regarding the effects of handcoding with respect to interpretation of the model, I can now draw the following conclusion: the interpretation of encodings constituting the provision of content, or carrying content with the assumption of a content provider when no such provider exists, amounts to a *false measuring theory* — the interpreters of the model may attribute content to the model in virtue of those stand-ins, but such attribution does not accurately capture what in fact the model demonstrates. Researchers that assume this false measuring theory are then subsequently led to have to handcode new representational “token place-markers” (or handcode the generative procedure for their potential production — and thus, all the possible combinations of “token place-markers”) in the construction of their models. This exposes the relation measuring theories have with theoretical model construction, and thus the limitations false measuring theories impose: without recognition of the falsity of the measuring theory, the research programme is forced to work within the limits of framework assumed — if the phenomena in fact extends beyond those limits, the research programme is incapable of accounting for them. (A product of an encodingist research program — an actual model constructed — may in fact instantiate real representation that can emerge and change, but this is likely to not be recognized by the researchers in the program as such, and cannot be attributed as a success within that research program.)

Second, Bickhard & Terveen (1995)¹⁰⁸ present an extensive review of current cognitive science and AI programmes and positions and their convergences with and divergences from encodingist assumptions; one of their conclusions is that a predominant

¹⁰⁸ Based on additional analyses found in Bickhard (1980b, 1993), Bickhard & Richie (1983), and Campbell & Bickhard (1986).

number assume implicitly or explicitly the encodingist assumption. On the assumption that Bickhard & Terveen are correct that most of cognitive science and AI assume encodingist positions with respect to the nature of representation, it follows that the models produced by these programmes are handcoded with respect to representational content emergence and change. Conversely, identifications of existing handcoding that results in precluding fundamental content emergence and change stands as evidence of encodingism. What I'm offering here is an explanation for why the computational models I have reviewed in Chapter 3 are handcoded with respect to representational content emergence and change: it is because they are based solely on encodings — they assume an encodingist paradigm, in which encodings are posited as a logically independent form of representation that is sufficient for explaining the role of representation in analogy.

And, third, to reiterate, representational encodings do legitimately exist (for example, Morse code¹⁰⁹). They just depend on a more fundamental form of representation, the content of which may be used to provide representational content to an encoding stand-in. It follows that *some* phenomena can be legitimately modelled relying only on encodings, as long as that phenomena doesn't require an account of the *source* of content. I claim that creativity, learning and development, however, fundamentally depend on the possibility of emergence and change of representational content (and structures). This emergence and change amounts to the possibility of the agent representing (knowing, understanding, "seeing") something fundamentally different about the world than was possibly represented before (whether it's correct, of course, depends on verification).¹¹⁰ A paradigmatic example of the kind of phenomena that is integrally involved in creativity, learning, and development is analogical cognition. This is why I have taken it as my case study in this project: analogy is a cognitive process that involves this kind of emergence and change — a cognitive process that involves *conceptual change*. *Some* aspects of analogy might not involve emergence and change; to the extent that they don't, modeling with encodings may be appropriate. To the extent that they do, however, encodings are not sufficient. Using only encodings in a model precludes fundamental emergence and change, and, therefore, precludes a deep account of analogy — or any other creative, learning-based, or developmental phenomena. *Any result or behavior* that is normally a *product* of emergence or change has to be specifically anticipated in the encodings (or possible combinatorial combinations of encodings) — and these results

¹⁰⁹ For example, in the case of Morse code, I could present you with a rule that stipulates that '•••' means 'S'; subsequently when I present you with '•••' you know that it stands for — that it represents — 'S'. However, the reason why this works is because you already know what 'S' means. The association rule I presented you with serves as the epistemic contact you needed so that you know to apply the same content you do to 'S' to '•••'. In this sense, the rule I gave allowed for content transfer from 'S' to '•••'. But the stand-in relationship I provided in no way provides *new* content; encodings can only change the form of representation already available — they are logically dependent on a content provider. The key question that remains is: What is the nature of your initial representational content of 'S'? Clearly it does not exist in the external symbol 'S', but somehow is inside you-as-the-interpreter.

¹¹⁰ "New fundamental representation" can be viewed from an encoding perspective as a "new representational building-block"; but, as I will show in a moment, representation in interactivism is in terms of (ontologically emerges as properties of) aspects of system process organization, so the building-block metaphor does not extend all the way.

have to be specifically handcoded by the researcher and appropriately interpreted. However, we need an account of how those results were arrived at — how conceptual change via representation emergence or change occurred. Handcoding the results therefore assumes what needs to be explained; from the perspective of explaining these results, handcoding begs the question.

4.2 - Interactivism

In this section, I present an outline of *interactivism* (Bickhard, 1980b, 1993, in press; Bickhard & Richie, 1983; Bickhard & Terveen, 1995; Campbell & Bickhard, 1986). My goal here is to explain how it can account for the ontological emergence (and thus the possibility of temporal emergence) of representation — what encodingism fails to do. Interactivism, however, is not motivated solely by the desire to avoid encodingism.¹¹¹ It has been developed as a combination of several intellectual traditions whose roots have interesting similarities, but have not, until now, been integrated into a consistent framework. The traditions include Piaget’s constructivism, Gibson’s ecological theory of perception and affordances, and the pragmatist approach to knowledge. There is also convergence with the later Wittgenstein’s (1953) “meaning as use” theory of meaning. There are also, however, some important divergences from aspects of these traditions.¹¹²

As I concluded in Section 4.1, encodingism is fundamentally incoherent. In particular, the result of such incoherence is that the premise of encodingism — that representational content is ultimately *provided by* (or even, *the same as*) epistemic contact — is false. It is here that interactivism makes its break with encodingism: interactivism avoids the incoherences of encodingism by making a fundamental split between what constitutes representational content and epistemic contact.

However, leaving representational content (what it is in the representation that makes it *about* what it represents) *unrelated* to epistemic contact (whatever makes the functioning of a representation appropriate to the current situation in the environment) would be a mistake. This would leave us with a new dilemma: we wouldn’t have any epistemic access to the world via our representations and our content would constitute all there is of our world; representation would lose its connection with the world (we would end up in a strange place, somewhere between solipsism and idealism). We still want our representations to be about the world while also not having the representations constitute all that there is of the world. Interactivism thus explains their relation so that the notion of representing is kept intact.

¹¹¹ Historically, in fact, Bickhard developed the basic interactive model *before* he discovered the encodingism critique.

¹¹² Specific issues are dealt more comprehensively in the following: Bickhard (1980b, 1987, 1995), Bickhard & Campbell (1992), and Campbell & Bickhard (1992) particularly develop the interactive framework with respect to language; Bickhard & Richie (1983) investigates Gibson’s theory of perception and it’s relations to interactivism; Campbell & Bickhard (1986), and Bickhard (1988, 1992a,b) examine Piaget’s constructivism and develop the interactive model in development; and finally, Bickhard (1993) and Bickhard & Terveen (1995) particularly focusses on the move from rejection of encodingism to adoption of interactivism, as well as identifications of encodingism in current cognitive science and AI projects.

The next move that interactivism makes, related to the fundamental rejection of encodings, is to choose a particular kind of ontology for representations — one that allows for emergence: a process ontology. This is in contrast to encodingism's basic substance ontology: while encodingism focusses on the *elements* of representation (as per a substance ontology), interactivism requires a shift to a view of representation as being a *functional aspect of certain sorts of system processing*. This shift to a *functional* model of representation means that interactivism provides an explanation of *representation* (*how* something represents or is representing), rather than *representations* as (fundamentally) things (although from the functional perspective we can make reference to the functional organization of a system which would participate in representing, as I will do below).

In order to explain this model of representation as being a functional aspect of sorts of system processing, I will first present the high-level description of interactive representation, followed by a more detailed analysis. Interactivism starts with this initial orientation: "In its broadest sense, the only function that a representation could serve internal to a system is to select, differentiate, the system's further internal activities" (Bickhard & Terveen, 1995, p.58). This constitutes the basic locus of representational function. Two additional, complementary logical requirements must be accounted for, however, if this representational function is to meet the conditions required for full-blown cognitive representation; these also necessitate the distinctive features of interactive representation.

The first logical necessity is the *possibility of error* (the third property of representation outlined in Section 3 of this Chapter). The need for the possibility of error arises because the functional differentiation of the system activities must be in some sense epistemically related to some environment being represented — that is, there has to be some sense in which the system's functional representing of the environment could possibly be wrong. If a representation was never possibly wrong, that would mean that the system directly knows the environment. This claim does seem to serve as its own *reductio*, as any epistemic system clearly does not have such omniscient access.¹¹³ Representations constitute the system's model of the world, and for the system to be able to tell the difference between its model of the world and the actual world, it has to have some way of *functionally detecting* that its model (its representation) could be wrong.¹¹⁴

The need for the possibility of error, in turn, requires that the system be able to act and interact with its environment. This is required because without such action (without

¹¹³ Note that this is not to be confused with the surface similarity of Gibson's "direct perception" — for one, perception is not the same as knowledge.

¹¹⁴ As Bickhard & Terveen (1995) note, the criteria of simply being wrong is too weak as it allows any observer semantics to determine such "wrongness" and thus yields a semantics for that observer, but not for the system itself. This parallels the problem I raised above in Section 4.1 about requiring an observer to provide representational content — the problem here is that only the observer would know in what sense a representation was wrong. If this were necessary to explain error, then a necessary condition for representation would be explicated in terms of already representing observers — this results in a vicious circularity.

an output) and some kind of subsequent uptake¹¹⁵ of the results of that action, the system has no possibility of determining the entailments of what it thinks the world is like based on its representation of the world. A strictly passive information processor has no way of checking such representations against the world. With an output and some kind of uptake of the results of that output, the system now has a connection between its action based on how it represented the world as being, and how the world actually is.

Simply having output and subsequent uptake of the results, however, is still not sufficient for the system to be able to detect any possible error. This leads to the second (complementary) logical necessity: that *this error must be for the system itself*. To satisfy this requirement, those differentiations must in some sense constitute at least implicit predications about the world that could be true or false *from the perspective of the system itself*.

There are two levels that I will consider at which this error detection via implicit predication from the system's perspective is possible. At the base level, all that is required is that the system have a functional link between some internal state and an internal action-generating subroutine (this subroutine, if engaged, would carry out a process that would somehow affect the environment — i.e., provide an “output” to the environment) paired with another internal state that the system should result in if the action-generating subroutine is engaged. Which actual outcome state the system in fact ends up in after selecting an action will depend upon the results of the action on the world, and how the world, in turn, affects the system (this is the uptake). This functional link that serves as a pointer from one internal state to another via some action is an *indicator*. This indicator serves to implicitly predicate of the world that the world is such that if the action is taken (i.e., the internal action-generating process is selected), then the expected outcome should result. It is *implicit* predication because the system itself does not directly know what the actual environmental conditions in fact are; rather, it only knows that the conditions are such that a certain action is expected to arrive at a certain internal outcome. It is this implicit predication, however, that could be true or false (i.e., that has a truth value) from the system's perspective: whether or not the indicated outcome is in fact reached is a system detectable condition — a purely functionally detectable condition — and failure to reach the indicated outcome falsifies the indicator (and the uptake serves as error feedback) (Bickhard, 1993; Bickhard & Campbell, 1996; and Bickhard & Terveen, 1995).

Despite the capacity of the first level to achieve system detectable error, it is not enough to account for how the system might make use of the fact that its indicator was falsified. Currently, at the first level, the possibility of functional detection of failure of an indicator (with its implicit predication) is just there — it has no specialized consequences. What is lacking is a way for the system to turn this error outcome into information that could be used for additional processing (e.g., to reiterate the interaction, try a different interaction, or invoke some learning procedure). The way to add this functionality to the system is to add some notion of *goal-directedness* — some reason for

¹¹⁵ In the sense that the environment causes the system to result in some subsequent internal state — *not* that some kind of information was subsequently input to the system, otherwise we have the same encoding problem (and therefore necessary reliance on handcoding) all over again. I will address this ‘uptake’ in differentiation, below.

why a failure to be correct about the world has entailments for the system itself: because the system, due to the error, did not reach its goal. This is the second level. This gives the system its own internal criteria for setting correctness¹¹⁶: “*The logical function that goals serve in the interactive model is to provide criteria for error*” (Bickhard & Terveen, 1995, p.63). While a goal may in turn be representational, this is not a requirement — otherwise representation would be explicated in terms of representation. Instead, “goals need only be internal functional switches that, for example, switch back into a trial and error interactive process or to a learning process under some conditions (functional failure), and switch into further processing in the system under other conditions (functional success)” (Bickhard & Terveen, 1995, p.62; see Bickhard, 1993, for more detail). Since living systems in general will not detect error unless they can do something with the error information, I will, from here on, assume this second level of functional system organization in my presentation of interactive representation (put another way, the first level is likely to be irrelevant when considering systems that have evolved and whose successful continuation to exist depends on survival in environment).

With the stipulation of action and interaction, in combination with the requirement that the error be from the perspective of the system (and thus, that action selection based on what a representation indicates for the system is in the context of a goal, in order for that error information to be used for further processing), an action can then be intuitively thought of as being like an empirical test for the system itself:

“I think the world is such that I can do X . If I am correct (i.e., if my representation, which indicates that the world allows for X , is correct), then I will in fact be able to do X . But until I actually try to do it, I will not know if the world does allow me to do X . If I can’t do X , then I have to revise my representation of the world.”

This sketch sets the foundation for how the system, through interaction with the environment, extracts the information to shape and modify its own behavior, and even modify its own representation (by changing its internal functional organization), relative to how the environment is. This, in turn, shows how the function of representation closes the circle from internal system function of what an action should produce, to action, and then to the system’s perspective of the outcome of the action. And this is where interactivism gets its name — *interactivism*. I will refer to this system functional organization that openly requires contribution from both internal functional indication of the system and resultant environmental feedback as the *functional interactive loop of representation*. Figure 4.1 is an incomplete sketch of this cycle (the particulars of Figure 4.1 will be filled in what follows):

¹¹⁶ See Bickhard & Campbell (1996) for more detailed elaboration of the necessity of and requirements for the possibility of system detectable representational error for learning and development.

Figure 4.1 - The Cycle of *Interaction* — the *functional interactive loop*

The outline thus far presents the basic intuition of the interactive view of representation as *anticipation of interactive potentialities* (that is, anticipation of potential actions and interactions are possible). Accounting for how these “anticipations” are implemented in the functional organization of a system, however, requires more discussion, and is the task which I turn to now. In doing so, I will now fill in some more of the details concerning the particulars of interactive representation and how it can be considered representation: i.e., what constitutes epistemic contact, what constitutes representational content, their relation, and how representational content ontologically emerges (and therefore may possibly temporally emerge).

First I will present how Bickhard (1993; Bickhard & Terveen, 1995) explains epistemic contact. Consider a system or subsystem in interaction with an environment. As noted, the course of the interaction will depend in part upon the internal organization of the system itself, but in part it will also depend upon the environment being interacted with. Thus, differing environments may yield differing flows of interaction, and differing environments will likewise leave that (sub)system in differing final internal states or conditions when the interaction is “completed.” These possible internal final states will serve to *differentiate* possible environments. For example, environments that yield internal final state **S13** will be differentiated from those environments which yield internal final state **S120**. These possible final states, in turn, will *implicitly define* the class of environments that would yield that state if in fact encountered in an interaction. This differentiation and implicit definition is what constitutes *epistemic contact*.

While this differentiation and implicit definition by a final state constitutes epistemic contact (reaching of the final state is, in part, contingent on the conditions in the environment), the final state itself does not indicate anything at all about its implicitly defined environments — except that they would yield that final state. Again, this is where interactivism diverges from encodingism: “A possible final state will be in *factual* correspondence with one of its implicitly defined environments whenever that state is in fact reached as a final state, but the state per se contains no information about what that correspondence is with — the relationship to the corresponding class of environments is purely implicit. *Thus there is no semantic information, no representational content,*

available that could make that final state an encoding” (Bickhard & Terveen, 1995, p.60).¹¹⁷

These possible final states constitute a basic representational *function* without themselves bearing any representational content (nothing is represented about the implicitly defined class of environments except that it is different from the other differentiated classes). As Bickhard & Terveen (1995) point out, this seemingly small point is what makes interactivism invulnerable to the incoherence problem that encodingism lands in: “In particular, an interactive differentiating final state does not require that what is being represented be already known in order for it to be represented. It is precisely that requirement for encodings that yields the incoherence of foundational encodings” (Bickhard & Terveen, 1995, p.61).

The next task is to specify how interactivism accounts for representational content, without providing it from the observer perspective as encodingism requires. In its most concise form, *representational content* is defined as *indications of potential further interactions*. This picture is made more clear by adding a little more detail to Figure 4.1, producing Figure 4.2.

Figure 4.2 - Representational Content as
Indications of Potential Further Interactions

¹¹⁷ Note, however, that (passive) differentiations (e.g., transductions) are precisely what are standardly considered to be encoded representations, as in the encodingist transducer semantics example I used in Section 4.1.

Now, consider a system which has some internal state that it is trying to reach — a goal state. This goal state serves as part of the mechanism for “choosing”¹¹⁸ what action to take next.

Next, consider that the system is currently organized such that if it is in a certain state, say n , then a variety of possible future interactions are indicated — in Figure 4.2, these indications are labeled α , β , and χ . These indications are pointers to possible action-generating subroutines (e.g., the process for the tensing of a particular muscle) that could be carried out. Along with the actions are also indicated the internal system states which should result given the action. In Figure 4.2, the functional indication a indicates that if the action is carried out, then the internal system state m is expected to result. This internal state m , in turn, functionally indicates further possible interactions, with associated action generation and expected outcomes (α_1 , α_2 , α_3) — these are the “further interactions.” In this way, the indicator α indicates potentials for further interaction (α_1 , α_2 , α_3 may likewise each indicate sets of new further potential interactions, and so on).

These networks of indications from given states in certain given conditions constitute potential system processing. Selection is then based on what the current goal is, and how that goal might be reached if a certain indication of future potential interaction is “chosen.” In this way, the system can use the differentiations in these indicated final states to differentiate the system’s own internal goal-directed processing. For example, if the system in Figure 4.2 is trying to reach a particular goal state, then under certain conditions, choosing the action indicated by α might be the path to take, while under other conditions the action indicated by β might be more appropriate (the conditions warrant choice of different potential future interactions).

These processing selection dependencies constitute representational content *about* the differentiated environmental classes. Thus, in the example from Figure 4.2, if I am in internal state n , and, therefore, in the environmental state implicitly defined by n , then that environmental state is (implicitly) predicated (by the internal indicator — the dashed line α leaving state n) to have the interactive properties that would yield internal state m (and, therefore, the environmental state implicitly defined by m) if I should engage in interaction α . (Similarly, being in internal state n is also predicated to have interactive properties that would yield other states if I engaged in interaction β or χ .) Here, the representational contents are the possible selection-of-further-processing uses that can be made of the differentiating final states (e.g., the expected state m) of a chosen interaction. Conversely, the final states and their indicators indicate the further interactive properties appropriate to whatever selections of further interaction might be made on the basis of those final states.

¹¹⁸ The word “chooses” is quoted to highlight that it is *not* intended to be taken in the sense of *necessarily* requiring some homunculus which then “decides.” Rather, a choice could instead be a matter of what path the system *will* take when certain conditions hold (perhaps set by the goal state and other possible conditions). They are choices only in the sense that when we remove considerations of what particular conditions currently hold and look to see what the different kinds of possible actions are that could be taken in all kinds of different conditions, we see that there are multiple possible paths — and as will become clear shortly, these paths are indexed according kinds of conditions which make those paths appropriate relative to the goal. “Choices” could also, of course, be stochastic.

Although epistemic contact plays a different representational function than representational content, the two are necessarily related, and this relation is part of what constitutes the capacity for the system to functionally detect representational error and to, subsequently, potentially modify its current functional organization: if an indication outcome is not met — if what was expected to happen doesn't — then the system has available to it the information that an error has occurred. For example, in Figure 4.2, if outcome state *o* is not the same as (or close enough to) the expected internal system state *m*, then the system may ratify its indication links. In this way, the state *m* has a truth value for the system (it is true if outcome state *o* matches *m*, and false if they don't match), and this truth value is contingent on the actual environment. So representational contents, through interaction, depend upon the outcomes which are a result of what the environment is in fact like — but the outcomes, for the system, are in terms relative to what was expected.

This possibility for error, in turn, can be used for functionally distinguishing when processes for changing internal system organization, such as learning, should take place. For example, if the expected state *m* does not match the outcome *o*, then the system may change its internal organization links so that in the future when in internal state *n*, amongst the possible future interactions indicated will be a link to state *o* (and whatever possible future interaction links follow from it). This would constitute *new* representational content (and is a case of a possible process for temporal emergence). A number of possible learning mechanisms could be posited now that the necessary condition of having the possibility of error which is system detectable has been satisfied. (Very important: any learning that does take place will be in terms of changes to internal system functional organization.)

Now a complete account of the ontological emergence of representational content out of the functional organization of system processes is possible. First, note that a model of the emergence of a functional process (like representation) must be *independent* of issues of representation because function is logically prior to representation; the ontological emergence of representation is thus modeled *within* that framework of functional process (Bickhard, 1993; Bickhard & Terveen, 1995, p.93). Interactivism satisfies this constraint because, as is clear from above, all discussion of the functional composition of a system has been done without assuming the existence of already fully representing components; rather, the components are then built out of these system functions. And this is what gives interactivism the capacity to account for the ontological emergence of representation out of what is non-representational (the components of system functional organization).

The possibility of *temporal* emergence (based on the explanation of ontological emergence) is then found in the potential for processes of construction of new possible indications linked with new implicit definitions (new environmental differentiations) — that is, an internal system process may construct a new functional link between some

internal system state and a part of the system for producing an action¹¹⁹, along with its associated expected outcome state. This possible construction is afforded precisely because of the split between representational content (which is now defined as internal indications of future interactions) and epistemic contact (which is defined as the environmental differentiation class that the internal differentiation-state implicitly defines). Such new indications can be constructed out of initially non-representational, functional components, unlike the case with encodings. Possible combinations of what is implicitly defined and what interactions are afforded are constructed by contributions from the system *and* from the environment — in no way is there the need for *already representational* “primitives.”

“Legitimate” encodings, from the interactive perspective, are now seen as the following: what assigns an encoding’s representational content is a property of the functional usage of the encoding by the system — it is a property of the system knowing what the encoding is supposed to represent — *and not a property of the encoding element itself*. In this sense, interactivism is a more fundamental form of representation than encodings in that it is possible to construct *derivative* encodings on an interactive, functional representation base, but not the other way around: “[interactivism] provides an account of the ‘ground’ or ‘foundation’ for representational content that encodingism cannot” (Bickhard & Terveen, 1995, pp.56-57). Thus, interactivism has the potential to account for all of the phenomena which involve derivative encoding representations. This, in turn, includes the representational grammar thus far considered.

Certainly an interactive foundation could support the grammar in terms of stand-ins for interactive indications of further interaction, as well as support the rules for the functional relations between the stand-in encoding units of the grammar (e.g., what other units a given unit is associated with, and under what indicated conditions). However, for this grammar to exist, it must rest on top of this ground level of interactive functional organization. Furthermore, existing grammar components may be added, removed or changed only (but at least possibly) by making the requisite changes to the underlying interactive grounding. I will address some additional points regarding the nature of representational grammar in Section 6; here, the conclusion is simply that such a grammar is possible *if* interactively supported.

Bickhard & Terveen (1995) present a nice summary of the three core aspects of interactive representations (p.92):

- (1) *Epistemic contact* — interactions with the environment terminate in one of two or more possible internal final states, thus implicitly differentiating the environment with respect to those possible final states. This is the *epistemic*

¹¹⁹ Note that interactivism does not require that the internal subsystems for the generation of “action” be literal bodily actions which affect the environment — they may be movement of perceptual apparatus to orient the system to particular sensory stimuli (e.g., saccading the eye in a direction to see a visual object more clearly). This kind of action doesn’t *affect* the environment itself (at least not much). The actions may even be “internal” to the system — for example, waiting for something to happen (the interaction is in terms of remaining still but expecting the same internal state to result; Bickhard & Richie, 1983, discusses this in more detail), or doing an internal activity like mentally imaging an event, etc. Internal or “mental” “interactions” will clearly be different from external interactions in terms of components and what is interacted with — but the same functional interactive loop remains. I will return to this point below.

contact aspect of representation — the manner in which interactive representations make contact with particular environments.

- (2) *Functional aspect* — internal states or indicators, generally constructed with respect to dependencies on such final states, influence further system processing — this is the *functional* aspect of representation and is the only role representations can play within a system.
- (3) *Representational content* — through influencing goal-directed interaction, which either succeeds or fails in achieving its goals, *representational content* emerges in the organization and functioning of a system as falsifiable implicit interactive predications about the environment — representational content has a truth value that is fallibly determinable by the system itself, not just by an observer.

This outline which I have presented is only the account of how interactive representations can capture the minimal requirements for being representations and how they avoid the problems of encodingism (the deepest being an account of representational content emergence). However, there are clearly a number of issues that remain to be addressed — many of these being the proper subject of a whole research program. I now turn to present the core perspective of the interactive approach to representation in the context of an account of ascending to more and more structurally complex representation that is inherently *situated* in its emergent level context.

5 - Situated Representation

With the outline of interactive representation above, I have shown how representation ontologically emerges. I have also described the *potential* for how processes could construct new functional system organizations (or change existing organization), resulting in new interactive representation system organization (the temporal emergence of interactive representation), through error indication in the context of goal-directedness. Yet to be addressed, however, is in what sense this representation might start to acquire what I have up to this point been referring to as “*structural complexity*.” The starting point is to conceive of “representation structure” as internal functional organizations of system actions paired with outcome differentiators that allow the system to interact successfully with complex interactive properties of the environment. My proposal for how such structure is acquired is to conceive of the increase of internal representational power (to differentiate increasingly complex external structure of the environment) from the perspective of a successive hierarchy of *situated* interactive representation — *situated representation* — explained as follows.

Following the core ontology of interactivism, representations aren't things, but rather, processes — specifically, processes consisting of the functional interactive loop defined above (Section 4.2). *Situated representation* are these functional interactive loop processes (of interaction, manipulation,

and change) arranged in a hierarchy, where the processes at level $n+1$ interact with, and potentially create new, or manipulate and change existing, processes at level n . I characterize this by saying that each level is *situated* relative to the level below it. So a consequence of my view is that the functional interactive loop does not have to interact directly with the environment external to the organism. Rather, each level interacts with, *and hence is situated with respect to*, a level below it in the hierarchy. The final base (or “bottom”) level just happens to be situated relative to the system’s external environment (constructively, this is also where the hierarchy begins). It is the central idea of being situated that unifies all the levels. Finally, on my view, any constructed representational structure (any capacity to build functional interactive loops that can interact with increasing structural complexity of the environment) historically depends on the levels below it, and emerges from *ALL* of the levels working concurrently. This kind of emergence constitutes ontological emergence (derived from the interactive account above) of situated representational structure.

The above paragraph succinctly captures my theory of situated representation. But probably too succinctly. To really spell out the ideas in the above paragraph in detail, an example is needed. That is what I turn to now. In the grand tradition eloquently initiated in Valentino Braitenberg’s (1984) *Vehicles*, the example is a thought experiment in artificial life.¹²⁰

5.1 - The development of level 1: base indicator assemblies and chains

Imagine a solitary (virtual) creature or organism that lives in a (virtual) environment — a (virtual) pond say (from here on out, I’ll drop the word “virtual”). This creature needs to eat, and hence seeks out food. It lives in a mildly unfriendly environment: not everything in its environment (the pond) is good for it — is food, in other words. Some things in its environment are not only not food, but are not good for it to eat: they make the creature more hungry (this is not so biologically plausible, but this keeps matters simpler).¹²¹ The environment is only mildly unfriendly because it contains no predators — except for the problem of finding food, the creature is perfectly safe. When its hunger level gets high enough, however, it dies.

Imagine that the creature has some standard body parts such as a mouth and some way of propelling itself around the pond. It also has one feeler that is also a grabber. It uses its feeler-grabber to examine objects and grab the ones it wants to convey to its mouth to eat. The feeler-grabber cannot in one shot feel an entire object and determine whether it is food or not. This is key. The objects which are food or not are bigger than the feeler-grabber, so the creature has to feel around the object and swim around it too to

¹²⁰ This example is inspired by the DogPaste Project (Dietrich & Beyer, 1998).

¹²¹ This is reminiscent of the “subtraction stew” from Norton Juster’s wonderful book, *The Phantom Tollbooth* (1988).

determine the object's shape. (Of course, now one might wonder how the creature can get such large objects in its mouth. Good question. I just assume that its mouth is sort of like a snake's: it can open very wide so large objects can be stuffed in. This is the beauty of a thought experiment, after all.)

The creature has some additional properties. As time passes, the creature gets hungrier and hungrier. Hunger serves as a general control mechanism that influences the kinds of choices the creature will make (i.e., the kind of internal processes that are initiated, and what it does based on certain internal process outcomes). An excellent example of this kind of control mechanism is the computational temperature used in the HLP models, which I discussed in Section 5.3 of Chapter 3. In this case, however, hunger increases or decreases as a function what it has eaten (and perhaps with the passage of time). The creature also has avoidance behavior. Finally, the creature can innately associate its behaviors (e.g., moving its feeler-grabber or eating) with experiences it has had on the basis of interaction with the world. This is just a form of memory.

The creature begins its life not knowing what things in its environment are food and what things aren't. It therefore has to learn to distinguish food from nonfood. It does this initially by eating them. Eating food makes the creature less hungry and eating nonfood makes it more hungry. Importantly, nonfood doesn't leave the creature at the same level of hunger as before it ate the nonfood — instead it increases hunger; hence, eating a lot of nonfood will eventually run the hunger level up quite high. There is a predesignated limit to how hungry the creature can become. If its hunger hits that limit, it dies.

After learning, food objects are distinguished by their shape. In other words, the creature learns to associate shape with whether in the past it got less or more hungry after eating a certain object. Thus the creature, over time, comes to eat only food and avoid nonfood and so gets less and less hungry (and we can imagine, more and more healthy). I will be concentrating on this ability to learn about food, and the creature's knowledge of food and possible food objects. I will therefore be concentrating on the information the creature extracts from the interactions it performs with its feeler-grabber and mouth (eating). I will ignore as much as possible other information the creature might get, for example, from its legs or whatever it uses to propel itself around with. This will force the thought experiment to be even more artificial than it would be otherwise, but it also makes the thought experiment tractable.

As part of its ability to learn, the creature is capable of special internal “bookkeeping” which keeps track of functional links between internal system states and potential system processes (e.g., to generate actions). In particular, the creature is able to construct and keep track of interactive indicators. I will use the following data-structure formalization, which I will refer to as an *indicator assembly*, to describe the internal functional system organization of an interactive indicator:

- (1) <initial differentiation, system process, expected outcome>

The first element, the initial differentiator, is a possible “current state” of the creature (or part of the creature, such as a state in its feeler-grabber). The second element is the initiation of a further system process, such as generating an action (e.g., grabbing or eating) or initiating learning processes. And the expected outcome is a further internal-

to-the-system state — another internal state differentiator. With the creature so-organized internally, it is capable of functional interactive indication. This works as follows: when the creature is in the initial differentiator state, then the functional organization of the indicator assembly indicates that *if* the creature were to initiate the given action, then it will expect to find itself in the indicated internal outcome state (the expected outcome).

The *internal* specification of the interactive indicator by the indicator assembly (a state with a link to another next state via some action/process) is similar a finite state automata. When considering the functionality of the whole interactive indicator, however, there is a very important difference. In particular, the second element in describing an *interactive* indicator — the system process — produces changes in the environment (e.g., acting in the world), which in turn causes (or would cause, if the indicator is correct) the resultant internal state (outcome differentiator) in the creature. The indicator assembly described is therefore *not* a finite state automata in the classic sense of a passive system waiting for external input to move it to its next internal state; nor is the link from one state to another simply a ‘state-transition’: it is the carrying out of some action-generating process. This is a critical distinction which drives home the point that *all of the indicator assemblies are situated*: they are situated relative to some environment, depending on what level they are at — at the lowest level (level 1) indicator assemblies are situated relative to the outside environment; at higher levels, they are situated relative to the next lower level, and hence relative to the environment of the internal functional organization of the system itself. *And*, when the system does in fact go into the process indicated, it is *interacting* with the environment of its situated context.

The creature starts life with the innate ability to construct these indicator assemblies. How this construction works is best described through an example. Imagine that the creature is feeling some object. By interacting with that object, the creature will construct an indicator assembly. Of course, to start this, the creature has to tell the difference between what’s an object and what’s not. I assume that this is also innate to the creature (a gift of evolution): an innate ability to differentiate objects from pond-bottom (this, of course, could easily be learned through trial and error, but is a needless wrinkle for my purposes here). Objects consist of various patterns of stars and bars (*, |), and plain pond-bottom is the hash mark (#).

In the creature’s feeler-grabber, there are specialized differentiator states which the feeler-grabber will go into if it feels a star or a bar or a hash mark. State1 is triggered with star, State2 is triggered with bar, and State3 is triggered with hash. These states are different internal state differentiators. But further up in the creature, in its central ganglion, the internal states need to be unique so that, e.g., the object * | can be differentiated from ** | |. I do this by time-stamping the states from the feeler-grabber (time-stamping States1, 2, and 3). This means that the central ganglion internal state of the creature upon encountering a star is: {State1, <unique-time-stamp>}. However, I will suppress this complication in the following whenever context makes my meaning clear. (On the page, each triplet will be in a *different* spot — e.g., in the string of triplets in (6), below — which can be interpreted as the time stamp.)

When a hash (pond-bottom) is indicated, the creature automatically keeps moving (the creature is functionally set so that hash indicates the possibility of more “general wandering”). But when it encounters a star or bar, it starts an internal process of

constructing indicator assemblies. For example, suppose the creature comes across this object for the first time and its feeler-grabber is placed on the leftmost star:

(2) (Object1): *||****||****||*
 \diamond

(Object1 is the name I will use to refer to this particular object; the diamond (\diamond) indicates the position of the feeler-grabber.)

An example of an indicator assembly construction is then as follows:

(3) $\langle 1, R, ? \rangle$

(For convenience, I use 1, 2, 3, to stand for States1, 2, 3, respectively.) The 1 indicates the current differentiator state of the feeler-grabber. The question mark indicates that the creature has no outcome state expectations yet. And, the R indicates that the creature's next move is to the right (our right). The creature wouldn't have to do this — it could move left and feel the pond-bottom; it just happens to choose to move right. Upon feeling the next constituent of the object to the right, the creature then forms another indicator assembly:

(4) $\langle 2, R, ? \rangle$,

and then fills in the indicator assembly from (3) — because 2 happened to be the outcome state of moving to the right — to produce:

(5) $\langle 1, R, 2 \rangle$,

for a total of two indicator assemblies.

Learning how to differentiate what a particular structure “looks like” thus amounts to collecting the set of indicator assemblies that define the object currently being interacted with. In the case of Object1, the set of indicator assemblies thus formed is as follows:

(6) $\langle 1, R, 2 \rangle$, $\langle 2, R, 2 \rangle$, $\langle 2, R, 1 \rangle$, $\langle 1, R, 1 \rangle$, $\langle 1, R, 1 \rangle$, $\langle 1, R, 1 \rangle$,
 $\langle 1, R, 2 \rangle$, $\langle 2, R, 2 \rangle$, $\langle 2, R, 1 \rangle$, $\langle 1, R, 1 \rangle$, $\langle 1, R, 1 \rangle$, $\langle 1, R, 1 \rangle$,
 $\langle 1, R, 2 \rangle$, $\langle 2, R, 2 \rangle$, $\langle 2, R, 1 \rangle$, $\langle 1, R, 3 \rangle$.

During the interaction across the object, with indicator assembly construction going on concurrently, the outcome of each individual indicator assembly becomes the initial differentiator of the next indicator assembly, thus automatically linking the chain together. The 3 indicates the end of the object (reaching pond-bottom again), and thus indicates when the construction stops. Internally, the system now has a chain of linked indicator assemblies. I will call such a chain an *assembly chain*.

Note that if the creature had started on the right of the object and felt left, the very same assembly chain would have been constructed. Now let's consider what would happen if, after constructing the assembly chain in (6) by interacting with Object1, the creature encountered another object similar to, but not identical to Object1 — Object2, which looks like this:

(7) (Object2): *||****||****||**

Now, the final 3 expected of the assembly chain (6) would not be satisfied. Hence, the creature would have functional information — via the falsification of the last indicator assembly (assuming the creature happens to be interacting from left to right again) of the assembly chain for Object1 (6) — that it wasn't interacting with the same object felt before. This would then cause (as another feature of the creature's learning processes) the creature to construct a *different* assembly chain.

Now for some additional complexity. First, let's stipulate that all objects like Object1 are food; this is a brute fact of the world; let's also assume that Object2 is nonfood). How does the creature represent this? So far, at level 1, all it "knows" is that in front of it are certain differentiator patterns given certain interactions. Imagine that Object1 is the first object the creature has encountered. Let's imagine that the creature is also hungry. Upon encountering the end-of-object indicator of pond-bottom, the creature eats Object1 — it initially does this automatically because it is hungry and it currently doesn't have any knowledge of what's food and what's not. In keeping with the design of the creature, this eating is controlled by an indicator assembly:

(8) <?,eat,less hungry>

The question mark here is a system state reached after *any* object has just been differentiated; thus, the creature, at this point, automatically eats whatever object (any star and/or bar pattern, versus pond-bottom) it has just finished interacting with. While (8) is still just as much an indicator assembly, I'm going to call it an *eating indicator*. (Avoidance indicators are very similar in form to eating indicators, except the action is avoiding rather than eating; I will make use of these below, in Section 5.2.) At this point, even though the creature internally distinguishes between different objects (via its assembly chains), it does not take them to be food or nonfood. All objects, for the creature, simply afford the possibility of eating (by indicator assembly (8)), regardless of their consequences. Note that the creature does expect to become less hungry afterwards (i.e., the state of its hunger should go down). Even if this is falsified, the creature, at this point, does not do anything more based on this falsification. It simply moves on. To make such use requires going to another level of representational sophistication, which I present in Section 5.2

At this level, however, there are all kinds of other complications I could add while keeping the objects as simple, linear sequences of stars and bars. For example, the indicator assemblies could each have a confidence factor associated with them (how likely the interaction outcome will result). Learning would then be a matter of increasing the confidence factors of the expectations. This is more psychologically realistic, but also not required at this time. I could also make the objects more complicated by adding different characters as possible components. This too is an unneeded complication, but one that would no doubt help one's intuitions. At this stage, I have all the complications that I need.

Finally, just to stress the point, note that the assembly chain in (6) exists now at the lowest level. This is the level depicted in Figure 4.3 by the functionally linked chain of

indicator assemblies. Since each individual indicator assembly specifies a full functional interactive loop, the assembly contacts the world by the functional interactive loop “overlaid” into the external world — this specifies that the actual outcome of a potential chosen action is contingent on the world. In this case, the creature is depicted as actually successfully interacting with Object1 (i.e., I’m assuming the creature has already constructed the assembly chain for Object1 and is now interacting with it again).

Figure 4.3 - An Indicator Assembly for Successful Interaction with Object1

Two further notes about the figure: (a) the activity is assumed to be taking place over a sequence of interactions (from left to right), and thus, which state the creature’s feeler-grabber is in also correspondingly moves from left to right; (b) however, the functional *organization* of the feeler-grabber is characterized by the whole assembly chain.

5.2 - The development of level 2: chain differentiators

One could well imagine a simple creature equipped only with the apparatus described in Section 5.1. This creature would roam around in its environment; when it found an object (i.e., when it completed interacting with an object), it would just eat it; if its hunger went down, great; if it went up, too bad. Irrespective of such outcomes, and even potential information in the creature of what precipitated those outcomes, at this point it couldn’t learn from them — it can only learn whether it has come across some object before, or not. Such a creature as I have just described is even more simple than a paramecium, which can distinguish food from nonfood. And, in fact, my creature wouldn’t need to construct assembly chains. All it would need to do is bump into an object — any part of an object — and just eat it. But assembly chains are crucial to building differentiators at higher levels, and they make the creature more efficient. What I want the creature to do now is to somehow associate object identifying assembly chains

like the one in (6) (i.e., what in the world happens to be food) with eating, and to associate other chains (i.e., nonfood) with avoidance.

To achieve this, an additional function is added to the creature's learning (i.e., indicator assembly construction) processes: when the creature finishes differentiating an object (perhaps constructing an assembly chain, if the current interaction doesn't match what's been interacted with before), it then eats. Then, based on the outcome of that eating action (hunger going up or down), it builds a new eating assembly.

For example, imagine that the creature has just eaten Object1, the first object it encounters (i.e., the creature has built an assembly chain differentiating Object1, and then, based on (8), automatically eats). But, after eating, the creature is less hungry. Now the creature's learning processes will make use of the fact that the outcome of (8) was in fact reached — it constructs the following *new* eating indicator in (9):

```
(9) <( <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>,
      <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>,
      <1,R,2>, <2,R,2>, <2,R,1>, <1,R,3> ) ,
      eat ,
      less hungry >
```

This new indicator assembly is interesting because it highlights several important properties of situated representations at level 2 (and above). First, as noted in the initial description of the creature in Section 5.1, hunger works as a kind of control structure — in particular, it controls the flow of processing of the creature so that it will choose further courses of processing that will make it less hungry. Up to this point, however, there hasn't been anything in the creature to allow it to do this. Now that it can associate differentiated objects (by use of assembly chains) with being more or less hungry (by the actual outcomes of eating), it can build an eating indicator that in the future will allow the creature to expect future differentiations of the same type of object to result in less hunger. Lessening hunger, in this case, serves as an implicit goal of the system. Of course, the goal is explicit as a general control structure — but it is more importantly conceived of as implicit in the sense that the system itself in no way represents what *hunger* is or what would satisfy it.

This goal-directedness (at least in terms of hunger) did not play a role in the earlier constructions of level 1. Level 1 *did* have genuine interactive representation, but the creature's system could make no functional use of the potential falsifications of what was differentiated (except to build new assemblies that it hadn't interacted with before) — this is equivalent to the level of interactive representation without goals, as discussed in Section 4.2.

Now, in the context of the goal directedness to lessen hunger, the creature can build an indicator assembly to help it get closer to its goal in future interactions.

Second, differentiators automatically implicitly define a class of objects. For example, the differentiator in assembly (9) implicitly differentiates any object with the same interactive pattern as (6). Even if two different objects with the same pattern were interacted with, the creature could not tell the difference between them.

In the avoidance cases, the creature will create *avoidance assemblies* which will cause the creature to avoid eating objects that are nonfood (and hence help the creature avoid dying). Avoidance assemblies are just like eating assemblies but with opposite

actions. However, their expectations are different without being opposite. There is a small reward for avoiding an object, namely the creature's hunger stays at the same level as just before it avoided the object. Since over time the creature tends to get hungrier, avoiding an object stops this process momentarily and gives the creature some respite from increasing hunger. Not as much as eating food, but some relief, nevertheless. This works even if the object the creature is avoiding is food.

Avoidance assemblies work this way. Suppose the creature comes across Object2 (see (7), above). Object2 does *not* match the assembly chain in (9). At this point, we have choice in our design of the creature. We could stipulate that the creature still eats Object2 (it might be food, after all), or we could stipulate that the creature avoids eating Object2 on the grounds that it is not identical to Object1. The second choice is preferred because it is safer, but it also requires the naming process. Going with our second choice, and since Object2 is not identical to Object1, an avoidance structure is constructed (compare eating structure (9)):

(10) <(<1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,3>),
 avoid,
 same hunger level >

(The reason the avoidance assembly has the “same hunger level” as outcome is because the creature is mildly rewarded for avoiding something. The learning process that constructs avoidance assemblies takes care of this.)

To conclude section 5.2, eating assemblies like (9) (and avoidance assemblies like (10)) enable the creature to consider objects identical to Object1 as belonging to the same *type*. Now the creature functions as if it had the predicative concept: “objects of type Object1 are good to eat.” Of course, it doesn't actually explicitly think this, but we can loosely attribute this content to its eating structure in (9). We can consider this (i.e., constructing eating and avoidance assemblies) as existing at a *level 2*, the level just above level 1 where indicator assemblies are built and assembly chains are constructed.

5.3 - Name bindings and finding invariances: level 3 and beyond.

Now imagine what happens when objects of type Object1 start to disappear from the creature's environment. That is, imagine that creature has eaten so many objects of type Object1 that its environment is running out of such objects. What should the creature do? Its reluctance to try new things (like a child's) is over come and, let us imagine, the creature tries something new. Imagine that it tries Object2, which it has been avoiding all this time. And suppose that Object2 is also food. (For completeness sake, imagine that the creature has also tried several other objects and found them to be nonfood and so has avoidance assemblies built for them.) So the creature builds a new eating assembly similar to (9) but with the assembly chain for Object2 functioning as a name rather than the assembly chain for Object1, the assembly chain in (6):

(11) <(<1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,3>),
eat,
less hungry>

(Of course, now, avoidance assembly (10) has to be deleted from memory. A simple process can maintain such consistency merely by deleting avoidance assemblies whenever a new eating assembly is added to memory with an identically matching assembly chain. Note that eating assemblies never have to be deleted because the creature's environment is rather friendly: in the creature's world, food objects just happen to never become nonfood. However, some objects are avoided not because they have been sampled and found to be nonfood, but rather because the creature just has a tendency to avoid objects that don't match food objects.)

Now matters get more complicated. Suppose that there is another naming process, a more complicated one. This naming process supplies made-up internal *symbols* as names. So suppose that this processes names the assembly chain in the eating assembly in (9) G1. In other words, G1 is the name — for the creature — of the assembly chain in (9). However, in order for this name to work, G1 must be *bound* to the assembly chain for any examined object of type Object1. This binding is crucial to higher cognition (e.g., see Marcus, in press, and Markman and Dietrich, 1998). It is called *variable binding* because of its similarity to what computers do. Variable binding requires a new kind of organization maintained in memory. The organizations needed are order pairs of names and the assembly chain they are naming. Here is an example using G1 and the frame from (6):

(12) <G1, (<1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,3>) >

I call such organizations simply *name bindings*. Name bindings are internal encodings, as discussed in Section 4.2. These encodings, however, do not make the error of *encodingism* because these are all simply functional links between the assembly chain(s) and the names — the names themselves do *not* represent what they stand in for (the chain).

Crucial to name bindings are the innate processes in the creature that treat the first of the pair of items in the name binding (the one on the left) as the name, and the second of the pair (the one on the right) as the thing named. This is not question-begging. In fact, it is how (approximately) variable binding works in computers. (I should point out here that no data structure, even garden variety ones like stacks, ever by themselves carry the information they are intended to denote by the user. Rather, the data structure plus the processes for accessing it and manipulating it carry this information. For example, a list of numbers (1, 2, 3, 4) might be a stack, or an array, or a queue depending on whether, respectively, the numbers are added to and deleted from the list only on the left, anywhere, or added to the list on the right and deleted on the left.)

Now, equipped with the capacity to construct name bindings, the creature can construct even more abstract eating assemblies such as (13):

(13) <G1, eat, less hungry>.

Note that in (13), G1 functions as a bound variable, naming the assembly chain for some object. The eating assembly in (13) is even more abstract than the one in (9) because it can subsume *every edible object*. So for example, remember that Object2 is also food, and the creature now knows this (see (11)). Since Object2 is also food, it can be bound to G1:

(14) <G1, (<1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,3>)>

(A brief aside is needed to further explain how G1 works. Technically, G1 is the header of a list of assembly chains that denote food. Since the creature knows both objects of type Object1 and type Object2 are food, the name binding for G1 really looks like this:

(15) <G1, ((<1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,3>), (<1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,1>, <1,R,1>, <1,R,2>, <2,R,2>, <2,R,1>, <1,R,1>, <1,R,3>)) >

In short hand (not available to the creature), (15) can be rewritten like this:

(16) <G1, ((assembly chains identical to the assembly chain for Object1), (assembly chains identical to the assembly chain for Object2))>.

In short, G1 is now the name of a database of food assemblies.)

Now the question is, how does the system construct things like G1 in the first place? I claim that the key to situated representation is to notice that the problem of categorization is the same as the problem of representation (or more specifically, of particular differentiation that indicates particular further interactions). In the former problem, one has to place under one category, all the different instances of that category. For example, no two strawberries are identical, yet they are all strawberries. This is true for everything: all humans, all pipe wrenches, all sand dollars, all paramecia, etc. etc. (it's a big set). Representation faces the very same problem: one has to place under one heading — the name of the representation — all the different images or perceptions of that thing. For example, no two images of mine of our cat Asharu are the same. I see her now from this angle, now from that one. I hear her meow and jump around, I smell her after she tussles with a skunk, I touch her by patting her on the head, etc. etc. Yet all these perceptual images are of one thing — a cat named Asharu. The problem of constructing and maintaining my representation of her is just like the problem of

constructing and maintaining my category for strawberries even though I see and eat big ones, sweet ones, small ones, bitter ones.

Now we can see the importance of the name G1, above. I claim that names like G1, and name bindings, exist at the next level *above* name assembly chains, i.e., at level 3. Note that I haven't yet said *how* the creature knows to construct names like G1. In order to give our creature the ability to do this, I have to construct a *metaprocess* — that is, a naming process above the process for creating eating and avoiding assemblies that exists at level 2. I claim that this is a *logical* requirement: in order to get eating assemblies like (13), it is logically required that a process observe assemblies like (9) being created and use the information from such observations to construct assemblies like (13). Such a process, observing from above, as it were, is naturally called a “metaprocess” (see Dietrich, 1985).

Metaprocesses such as the one described just above might work by checking for and finding *invariances* and *differences* in indicator assemblies (and their components, which may include assembly chains). To see how this works, recall eating assemblies (9) and (11). The former used the assembly chain for Object1 (see (2)) and the latter used the assembly chain for Object2 (see (7)). Both of them have the same action and expectation. The two assemblies differ as to their precipitating condition — the assembly chain in their first position. When the two assemblies are executed, the same action is taken and the same result is achieved (namely, being less hungry) based on matching different assembly chains. The meta-naming-process at level 3 takes all this in as input in the form of snapshots of state changes (the states are readily available, remember, because all assemblies like (9) and (11) are, internally, finite-state automata that happen to indicate potential further system processing). The metaprocess in effect asks: “What do the execution of (9) and (11) have in common?” The metaprocess is hardwired to ask this, but it doesn't know the answer (also, the existence of the metaprocess is hardwired). The answer to this question is what was stated just above: both (9) and (11) wind up in the same state via the same action but from different starting states. The metaprocess “concludes” — i.e., arrives in its own outcome state denoting — that the differences between (9) and (11) are minimal and that their differences can be captured with a single name. In short, the meta-naming-process unifies assemblies like (9) and (11) on the basis of their expected interactive potential (expected outcome if certain actions are taken) and distinguishes them on the basis of their differentiators. This is different from the low-level individuation of interactive assemblies at level 1 (see Section 5.1). In this way, the metaprocess unifies at a higher level (level 3) what is disparate at the next lower level. This thus meets the condition I placed on representation above: that it plays a fundamental role in categorization.

With the metaprocess's output — namely, G1 — we now have a more or less stable bound variable, constructed on the fly, whose denotation — namely, the assembly chains — are known to certain processes internal to the creature. This is the beginning of a situated representation.

The construction of G1 above is a very simple example of the power of internal system-process naming. The point of the naming process is that unifies under one name several different eating assemblies (or avoiding assemblies) that end up in the same outcome state. Even the simple example given above increases slightly the efficiency of the creature because now it knows that several different things are somehow the same

(e.g., they result in eating). Knowing this could do something simple like make memory use more efficient. But to really see the power of the naming process (the meta-naming process), consider this variation of it. Suppose that the creature had the ability to pick out features of the assembly chains themselves using invariances and differences. So for example suppose that the creature could notice that the two food objects it knows about, namely, objects like Object1 and Object2, always start with a State1 and end with a State1 (the object itself has a star on both ends; note that a bunch of other combinations are possible). Now, suppose that the creature's level 3 naming process could name those features. This would mean that it could name the indicator assemblies in the assembly chains. Equipped with this capacity, the creature now can treat the names as *shorthand* for food objects, and now it can decide to eat some new object which it has never seen before based on merely whether or not it puts the creature in a State1 at both the beginning and the end. In effect, the naming of features of (parts of) assembly chains gives the creature *heuristics* about how to behave — about what actions the world might afford. Now the creature has a rather potent kind of efficiency — efficiency that is not merely internal economy, but allows the creature to selectively pick up on interactive features of the world that are more relevant to the satisfaction of certain goals than other features.

Note that the heuristic of eating objects that start and end with a star (cause the creature to go into State1 on both ends) could be wrong. This is the power of the interactive approach: it stress the importance of failure and being able to glean information from such failure. It could be an accidental feature. Now the creature can drop into a mode called “variation and selection” (for general discussion of variation and selection in this sense, see Bickhard & Campbell, 1998b; Bickhard & Campbell, 1996; Campbell & Bickhard, 1986) where it names different combinations of features and uses them until they fail; those names that are successful are retained, and may in turn serve as the basis for more complex names that include them as components in further variation and selection; those that fail might be modified for future attempts, or dropped altogether.

I claim that it is this latter kind of naming metaprocess that is crucial to the emergence of representation and learning about robust structured interactive potentialities in the environment.

Now for the final point. Notice that the metaprocess at level 3 is implicit: it is not explicitly represented in any sort of indicator assembly. Only its output is explicitly represented. This is true of all the levels and is, in fact, the general principle that is the key to this section:

Situated representations of increasing robustness emerge in a hierarchy of metaprocesses that make explicit the invariances and differences by focussing on certain features of the assemblies the processes of levels below construct. Picking such features might be a process of variation and selection.

I claim that looking for invariances and differences produces a hierarchy of increasingly abstract name bindings, i.e., variables that are bound to increasingly broad sets of actions and expectations. This hierarchy results in situated representations that are increasingly like the things we intuitively want to call structured mental representations. These

situated representations make more and more structure explicit in the underlying lower level processes and their outputs. In Table 2 (next page), I give a simple example of what I mean by using the creature we have been developing. For each level, I list the level's name (1, 2, 3, etc.), describe what it is doing, give an example of the indicator assembly it produces as output, and describe in quotation marks roughly how the process at that level interprets the behavior of the level immediately underneath it.

Level4	Some possibilities include noticing kinds of variations in the variation and selection process that worked particularly well (this would be going after process constructing strategies – i.e., learning strategies – that were successful). Further speculations will be given in Section 6, the conclusion.	

Level 3	Interactive assemblies with simple name-bindings	<G1, eat, less hungry>
	"the process at level 2 is building eating assemblies and all the relevant assembly chains start and end with State1"	

Level 2	interactive assemblies with assembly chains	<assembly chain1, eat, less hungry>
	"the process at level 1 is building frames and eating and avoiding"	

Level 1	simple interactive assemblies and assembly chains	<State1, R, State2>
	"interact with the world, differentiate"	

Table 2 - Ascending Levels of Representational Power

Note that how the higher levels are situated relative to lower levels with respect to the functional interactive loop can be complicated. For example, in Table 2, the G1 in level 3 is situated on top of level 2, but the actual action and subsequent expected outcome are at level 2. So the indicator assemblies at level 3 are hybrids: they can only exist at level 3 in terms of the name G1. But this isn't a necessity. An indicator assembly could exist solely at level 3. For example, consider an indicator assembly that takes a name (e.g., G1) as the initial state, but its indicated process was to *construct* an assembly at level 2, and its expected outcome is of the success of the newly constructed level 2 assembly reaching its goal (which, in turn, involves interaction with the world).

A major objection that might be raised at this point is that the representational capacity of the level constructing processes that I have specified above is subject to the same limitations of the representational grammar approach I criticized in the HLP models (the generative grammar that the Slipnet and codelet construction possibilities instantiate). This surface similarity glosses a critical difference between the two situations. The point is not that there are not constraints on what can be represented, there clearly are. Rather, what is innately constrained is *not* a product of base representational units, but on what grammars can be constructed. For example, Copycat starts out with a Slipnet and codelet grammar that is fixed for its entire life. The creature above, however, can *add* to its grammar components and change existing grammar capacities. It is this capacity that is necessary for creative analogy-making. I will return to this distinction again in the next section to make clear what kind of handcoding I have avoided, and what further handcoding I have not.

There remains the problem of how to account for the emergence of representations with *relational* structure. I take this up very briefly in section 6, pointing the way, rather than actually specifying how to do it.

6 - Conclusion

The creature in Section 5 really satisfies the constraints of interactivism. What it can know (true of interactivism in general) of the world are the potentialities for interaction afforded by the world. The creature predicates of these potentialities that certain further interactions will be successful, which in turn will help the creature achieve its goal. In this section I address some additional questions, challenges and remarks, and conclude with a summarization of what I have accomplished in this dissertation.

What about representation of relations?

As I demonstrated in Chapter 3, high level cognition, like analogy, requires some account of how relations in the world are represented. However, in the sketch of the creature given above, there is *no possibility* of the emergence of relational structure because it is simply not complicated enough. This is the beauty of this thought experiment: it is so simple that we can very carefully add capacities so that we get more and more robust representations emerging, all the while avoiding handcoding of representation (see below for further clarification); this is how I constructed level 3 from level 2. Now suppose we added a process to the creature that kept track of how long it

took to eat certain objects. For example, objects of type Object1 are rather long. Suppose that objects of type Object3 are also food and that they look like this:

(1) (Object3): **|.

Objects like Object3 are short. They don't take long to eat and therefore (by supposition) take less energy to eat. Suppose further that the creature can tell that such objects are just as nutritious as objects of types Object1 or Object2. Then imagine that it has the goal of minimizing energy spent for food ingested. Now the creature has a vested interest in eating objects of type Object3 rather than the longer objects of type Object1, and this could be reflected in additional control feedback, derived in part from basic hunger feedback. From here it seems possible that the creature would have enough information available to it to differentiate the relational structure "bigger-than": namely, keeping track of relative length of objects, and based on its goal of eating those that are shorter, shorter objects will be represented as better choices for eating than those that are longer (a differentiation within the food category). Furthermore, the representation of this relation would really be made distinct if the creature were faced with a choice between two equally nutritious (based on previous experience) food objects: two relatively large but different-sized objects would then have a "bigger-than" relation just as two relatively small objects would. This latter possibility would, again, be an extension beyond the creature's current perceptual capabilities — as of Section 5, it only views one object at a time, and then eats or avoids — but this is certainly a plausible extension. There is much more work to be done to provide a detailed example of representation of relations, but moving in this direction, I argue, is sufficiently plausible and will eventually lead to the kinds of relations used in the models I reviewed.¹²²

Representation of structural properties of the world

With the above thought experiment in Section 5, and the addition of the pointer to how relations might be accommodated, it is important to reiterate how the situated representation approach *can* adequately cash out the metaphor of "structured representation" that funds the use of the term (if not the actual implementation and subsequent entailments for the nature of representation) in the models of analogy I have reviewed. To reiterate a conclusion of Chapter 3, the metaphor of structure has been profitably used by the SMT and HLP models to account for how features of the world come to be represented in systematic and related ways so that processes involving comparisons of representations of these relations can lead to production of representations which demonstrate how situations have similarity on the basis of relations shared. I am not here going to provide a full account of analogical cognition, so I will not completely satisfy the challenge of providing an alternate account of representation that

¹²² I should additionally note that interactivism in general is amenable to Piaget's general constructivism, and to that extent, we can find further guidance in Piaget's discussion of how our familiar world of objects and events in space and time, including relations, can be constructed out of representations of patterns and properties of interactive potentialities (Piaget, 1954). Bickhard (1988) and Campbell & Bickhard (1986) provide a detailed discussion of similarities with interactivism, as well as crucial differences in Piaget's general approach to explaining development and developmental mechanisms.

includes its specific use in internal representation-comparison. However, as the thought-experiment creature demonstrates, situated representation does capture systematic representation of *structured properties of the environment* — a crucial necessary condition that must be met if any internal comparison of representation is going to be *about* the structure of the world.

For example, from an objective perspective of the environment the creature lives in, objects do have real patterns and real causal powers — i.e., real structural properties: Object1, for example, has the shape ‘*||****||****||*’ and is nourishing. This object thus has systematic interactive properties reflecting this structure: the first ‘*’ on either end of Object1 is adjacent to a ‘|’ and not another ‘*’; and if the object is eaten, it will cause hunger to be lessened, not increased (or unaffected). The creature can then come to represent these interactive properties by constructing internal indications of expected outcomes given certain actions. The systematic structure of the actual object is captured in the internal functional organization that pairs expectations with specific courses of internal system processing involving interaction with the environment.¹²³ As long as the creature has some way of differentiating the interactive properties of the environment, and outcomes of those interactions are relevant in some way for the creature’s goal-driven activity, then those properties may be represented, and as a result, the creature may build content about those properties (e.g., that feeling the pattern ‘*||****||****||*’ affords the possibility of eating to lessen hunger). Along with the sketch above for capturing relational properties, the situated representation framework should be able to handle accounting for environments that include relations as well.

How these representations might then be utilized for making similarity comparisons — for analogies — is still an open question. But *that* the situated representation framework can describe autonomous agent representation of situations involving entities and relations is not problematic.

Levels of situated representation: recursive complexity vs. knowing levels

An important distinction must also be noted regarding the “levels of situated representation” and “interaction between levels” that I have described thus far in my thought experiment explanation of situated representation. Up to this point, my discussion of levels of increasing representational complexity is best described as largely involving levels of the *recursive* use of the indicator assembly functional architecture (i.e., the functional interactive loop): the base indicator assembly that defines the core of

¹²³ But very important, the structure of the world is represented within the creature as patterns and paths of system processing, not as direct structural isomorphism between representational atoms that correspond to entities in the environment, along with relations between representational atoms corresponding to relations between entities in the environment. Rather, a situated representing agent only has to satisfy the constraint of being able to interact with structured properties of the world in the service of attaining certain goals; such representing may thus be functionally grounded in a wide variety of different internal functional system organization, rather than strict correspondence as structural isomorphism. Here, it is not the structural isomorphism (if any) that bears the weight of conferring representational status *about* aspects of the world, but instead indications of further interactions that may be useful to achieving internal goal-states. This is, again, an important departure from an encodingist assumption about what constitutes representationality — an assumption that seems to be clearly at work in current models of analogical cognition.

situated representation has been shown to be capable of increasingly complex functional organization through the recursive use of the indicator assembly functional architecture — e.g., linking indicator assemblies in chains, or collectively ‘naming’ (categorizing) such chains and using them as compound differentiators in new indicator assemblies. This recursive use allows for complex compound indicator assemblies — and with this complexity also comes the representational power to represent increasingly complex interactive properties of the environment.

These assemblies and their increased levels of recursive complexity clearly satisfy the definition of situated representation I provided at the beginning of Section 5: increasingly complex indicator assemblies are built in the context of interaction with an environment, which may include other indicator assemblies that may then be utilized (incorporated, manipulated or changed) in assembly construction. These more complex indicator assemblies require the antecedent construction of the indicator assemblies they make use of, and additionally constitute the capacity to represent more complex interactive properties of the environment (i.e., to differentiate environments that may afford certain further interactions, and to predicate further possible actions and outcomes of such differentiations). These more complex indicator assemblies exist at a “higher” level of recursive complexity (and therefore, representational complexity), dependant on the representational power of the indicator assemblies they make use of (or have changed) in the construction of the complex indicator assembly.

However, what about a whole level of indicator assemblies whose environment which they interact with is comprised solely of other indicator assemblies (that may, in turn, interact directly with the environment)? This latter sense of levels of interaction has been developed by Bickhard and Robert Campbell (Bickhard, 1978, 1980a, in press; Bickhard & Campbell, 1986), following the notion of *knowing levels*. These are defined as follows. Any given indicator assembly (or chain) will predicate interactive properties of an environment (and in this sense, may come to know how to interact with the environment successfully to reach certain outcomes). The indicator assembly itself, however, will have properties which it cannot itself differentiate and make predications of — that is, it will have properties that it cannot itself represent. Take, for example, the assembly chain that differentiates Object1. This chain does represent (differentiate and predicate — i.e., makes epistemic contact with and has content about) the interactive properties of the shape-pattern in the object. However, the assembly chain itself does not explicitly represent that this pattern is in fact symmetrical, or that there are three groups of double-bars (‘||’) (and many other features). Nonetheless, such information is implicitly present in the indicated path of system processing that involves a specific pattern of internal system state outcomes during interaction with the object. How might the creature make explicit use of such information? — how might it represent these properties of this ‘first’ level of interactive representation? It may do so through a second level of indicator assemblies that interact with the functional organizational properties of this first level. This second level might then come to differentiate patterns in the first level, and predicate of those patterns possible further internal system processing, including “internal actions” that may change the functional organization of the first level (perhaps the functional organization of assemblies that represent other objects that also have three sets of double-bars — e.g., other objects with three double-bars that were previously treated as non-food, but then changing them to indicate food), or lead to

actions in the environment. This differentiation with associated predication constitutes situated representation of the properties at this first level. I have already discussed a possible case of this kind of knowing-level interaction in the above example of how the creature might come to represent relations in terms of internal differentiation of interaction “length,” and predication of further internal processing on the basis of such length. See Bickhard & Richie (1983) for a similar example for rudimentary “counting” (a possible basis for the concept of number) on the basis of repeated use of first-level indicator assemblies.

In this way, these “knowing levels” of interactive representation are not just the recursive use of indicator assemblies in the construction of more complex indicator assemblies, but specifically involve indicator assemblies that interact with the properties of functional organization of other indicator assemblies. And, these naturally inherit the specification of *knowing* levels because such levels make explicit (by differentiating and predicating potential actions with expected outcomes) the functional information about the first level’s environment that is implicit in the *way* that first level interactively knows that environment, but does not itself functionally differentiate and predicate possibilities of further interaction. Also, these levels serve to give the agent as a whole potential representational access to its own manner of representing (a kind of self-reflection and self-reflective abstraction). Knowing levels are crucial for the representation of potentially quite complex abstract properties of the environment (see Bickhard, 1978, 1980b, and Campbell & Bickhard, 1986, for further discussion), and are also arguably crucial for robust psychological development (see also Bickhard, 1980a).¹²⁴

Knowing levels are likewise amenable to my general specification of the *situated* theme of situated representation because the representationality of the functional interactive loops (indicator assemblies) of these knowing levels must be situated with respect to their “internal” (sub-systemic — i.e., a system that is a component part of the general autonomous agent’s system) functional organization and the “environment” they interact with (which may be the other functional system organizations internal to the agent). In my definition of situated representation I have purposely blurred the distinction between *what* is interacted with in terms of knowing levels versus levels of recursive complexity in order to highlight how the situated notion is held in common for both levels of recursive use of the functional interactive loop and knowing levels of interactive representation: it is what makes both kinds of levels of representation in fact representational — how representational content and epistemic contact emerge and are functionally associated, and maintained or updated in that association, in the functional interactive loop to produce full-fledged representationality (complete representational aboutness with the possibility of system-detectable representational error). Nonetheless, the difference between levels of recursive complexity and knowing levels constitutes an important distinction because these kinds of system organization have entailments for the kinds of interactive properties of the environment that the system can represent — they constitute two directions of representational complexity and subsequent potential system

¹²⁴ There is also an interesting similarity between this notion of ascendance of knowing levels and Karmiloff-Smith’s (1992) account of *implicit* procedural knowledge being made *explicit* in higher-levels of representation through representational redescription (the *Representational Redescription* model of cognitive development).

knowledge.

Have I completely avoided handcoding representation emergence and change?

This discussion would not be complete without a final clear statement of how I have accomplished what I set out to do: to avoid handcoding representation so that I can account for the possibility of its emergence and change. Certainly, the situated representation account of representation emergence and change provides an interesting and novel approach to representation construction. But have I successfully provided an account that avoids the kind of handcoding I criticized the SMT and HLP models of committing? The answer is, of course, *yes*, but not without some important qualification. I have certainly dodged handcoding of the kind I have identified in the SMT and HLP models: handcoding that precludes any representational content emergence and change; but there are deeper representation-construction-related handcoding issues that I have not, and these must be distinguished from what I have done.

The distinction can be seen by first addressing this challenge (a variation of that addressed at the end of Section 5): As I have things now, every time the creature is going to build a new kind of representation beyond what it currently can (as above), a new process must be posited — i.e., hard-wired in. Doesn't this constitute possible handcoding?

Yes. But not in the sense of the presence of *any* representational content itself, as current analogy models require — avoiding this latter kind of handcoding has been the goal of the current project. Thus, I am handcoding the possible *kinds* of representations that *can* be constructed — of the kinds of representations that can emerge and change (this amounts to handcoding the capability to represent certain classes of environments in the world). But, it is not handcoding of the specific representational content — it is not handcoding that precludes *any* possible representational content emergence or change, as current models of analogy do.

To make this clearer, the challenge can be stated more pointedly:

It is agreed that the SME-based models (except, perhaps, Phineas) do not offer an interesting capacity for representation construction — the representations are simply input to the model (or LTM database/store), their identity antecedently set by their associated labeled units. But, the HLP models do offer representation *construction*. Specifically, the representational grammar analysis of Copycat uncovered that its representationality consists of pre-defined rules for building representational structures. It seems there is also a parallel in the creature's case, with its pre-defined processes for building recursively complex functional interactive loops — the essence of interactive representationality. From a constructive perspective, this seems no different than what Copycat faces: pre-defined rules for what can be combined, associated, broken, etc., in the building of representational structures — in the creature's case, it is the pre-defined rules for what states and functional links within the creature may be constructed when certain environments are encountered: Thus, the creature likewise has its

own representational grammar which it cannot break out of, extend or change.

There *is* a *big* difference even if there is a surface similarity in that both accounts involve construction. The crucial point (in the comparison between Copycat and my creature) is not that things are *constructed*, but in terms of what is or can be constructed. In Copycat, the atoms (e.g., the category of letter ‘A’) and the relations (e.g., ‘successor’) and any potentiality of their specific instantiation *have* to be antecedently determined and do not change — *and the full representationality of these components is fixed and complete* (i.e., nothing can be added or changed about that representationality). To use an illustrative metaphor, these constitute the “language” that Copycat has for representing its world. In an important sense, Copycat, and all of the other models reviewed, not only have a representational grammar (in terms of the rules for representation construction), but also have the complete set of possible representational units that the grammar will ever have available to work with — thus, metaphorically: the only “alphabet” or “words” (or “lexicon”) the grammar will ever have to work with is fixed (i.e., non-extensible and non-changeable).

The above challenge is misleading in that the real issue is precisely an issue of *representation* — of how content itself arises and plays a role in full-blown representationality¹²⁵ — not just an issue of construction per se. In the Copycat model, these atoms *are* the *only* possible atomic-content-bearing units about the world. In my creature, content can be any number of possible indications of potential further interactions¹²⁶, which may then be functionally paired with whatever interactive possibilities the world in fact turns out to offer (this follows directly from the specification of interactive representation constructive potential, as discussed in Section 4.2). Thus, what may be constructed *is* constrained, but the constraint is contributed by *both* the world (or, in the case of knowing levels, the “local” environment being interacted with) and the possible internal organizations of the agent — not *only* by the agent itself, as is the case with these other models: all the representational power about the world that may ever be applied, and the correspondence with what it may be applied to, already exists (and *must* already exist) antecedently within the Copycat model.

So, the proper comparison (again, relying on the “language of representation” metaphor) between Copycat and my creature is that with my creature, the possibility for building certain “alphabets” or “lexical” kinds is *constrained* — but *which* specific alphabets are built depends on the actual history of interaction with certain kinds in the environment, and on those intrinsic constraints of what kinds of internal organizations of the creature may be constructed or changed. Copycat (or any of the other analogy models) has *no* way of building a new variant of its representational “alphabet/lexicon,” and has no way of changing its current “alphabet/lexicon” (as just one aspect of this in-

¹²⁵ ... of where and how content meets contact with the world, and how the connection can be managed by the system’s own functional processes (hence, the important of system-detectable error)...

¹²⁶ Of course these are bounded: any combination of internal-to-the system states functionally indicated by links with any possible combination of actions, which, in turn, may be in any possible combination of variations of such chains, etc... But it seems legitimate to have such boundedness derive from the possible organizations of the system itself; after all, the claim is not that *any* system can represent *anything*.

principle constraint).

This is not to downplay that there is an important kind of handcoding that I have not addressed: the handcoding of these constructive processes themselves. This handcoding must likewise be cashed-out at some point because in-principle limits on the set of constructive processes seems to not allow for development that, at least in humans, is likely to occur naturally — and again, not just over phylogenetic evolution, but in ontogenetic development. Namely, it is likely that the constructive processes in babies are not simply the same as the constructive processes in adults.¹²⁷

An entailment of this is that there are in fact several distinct senses of the “grammar” terminology: the *representational* grammar of the SMT and HLP models includes the rules for application of representational content-bearing atoms corresponding to conditions in the environment; this entire grammar has been handcoded in these models, resulting in preclusion of the possibility of representation emergence and change. However, in the creature’s case, a new kind of grammar is handcoded that specifies the kinds of processes for the construction and manipulation of system functional organization — it is such organizations that may in turn be representational. But this latter grammar is a grammar of system organization construction, *not* rules for application of pre-determined and already assumed representational atoms.

Thus, I do still maintain that, even in the face of this handcoding, this is an important step removed from what even Copycat, with its proposal for construction, can account for, and concurrently provides a more coherent account of the nature of representation (an issue that to this point has not been adequately addressed in the analogy literature).

Summary of key results

Now to close. In this dissertation I have shown the following:

First, *I have shown what handcoding is and its effect on a robust example of computational cognitive science: explaining analogical cognition*. Handcoding involves the role that a model creator or interpreter plays in the creation or interpretation of a model. *Inappropriate* handcoding occurs when the creator or interpreter is somehow involved in the model’s functioning, or in the interpretation of the predictions made from the model’s structure or behavior, such that the model is deemed to provide results that match how the world itself is observed to behave, when in fact the creator ‘helped’ the model skip steps that the phenomena in nature to be explained requires, or the components of the model are interpreted as playing natural kind roles or instantiating kinds that they in fact aren’t. I developed a framework for the identification of inappropriate handcoding which makes clear *where* in the logic of explanation such handcoding causes damage, and subsequently, how to express *what* it is that has been handcoded. In the analogy models I investigated — models which are arguably representative of the variety within the general approach to representation taken by computational cognitive psychology — I identified present handcoding that precludes the

¹²⁷ And this is likely not just in terms of a primitive-to-complex sense (i.e., primitive processes being combined to create complex processes), but change in the primitive processes themselves, or adoption of new constructive process primitives, that adults have available.

capacity for representation emergence and change.

Second, *I have shown why handcoding is such a pervasive phenomena and why it won't go away easily*. At the root of the problematic existence of handcoding in computational cognitive science is the intimate involvement of the cognitive science researcher in the construction of complex computational models used to model cognitive phenomena — models whose complexity is required because of the complexity of the subject matter being studied. This involvement is more complicated and potentially problematic than any other science (old or new) because it is precisely what we (cognitive researchers) as intelligent humans do naturally that we wish to understand: how we think, reason, represent, and generally behave intelligently in the world. It is quite difficult, despite our best efforts, to not accidentally solve problems for our models which we take for granted or are unaware of solving, but that we ourselves must nonetheless solve in order to accomplish the intelligence-requiring feats that we do — what we really want is for our models to solve these problems *themselves*. Nowhere in computational cognitive modelling is this more clear than in the case of modelling representation. The mistaken tendency, as I have shown in the models I investigated, is to attribute representational power to systems which in fact don't possess true representation. This tendency, I believe, naturally arises because we, as observers and builders of these systems, can see the connection between what we construe as representational in the system, and what the system is intended to represent; and this is likewise true whether one is looking at the meaningful labeled units we have put into our machines and which are associated in ways that make sense to us, or whether we actually provide an environment (virtual or actual) that the system must maintain its correspondence with. The more difficult task is to account for how the system itself could represent — how the representations can be about something in the world *and* in what sense the system *itself* can make use of or draw information from this aboutness. *This* is the real problem that we wish to understand — the real problem that we ourselves (as autonomous epistemic agents) solve — and which we want our models to help us explain.

Third, *I have shown why the only way to avoid handcoding the emergence and change of representation is to avoid encodingism*. As I demonstrated, if we assumed an encodingist position, then any representation we posited in a model would necessarily have to be handcoded in that model: that is, any representation would have to be antecedently set to have some sort of correspondence with some aspect of the world, or interpreted as having some content about the world. Encodingism cannot, itself, provide such content — it can only borrow it. Unfortunately, our best models of analogical cognition, analyzed as representing according to a representational grammar, are most easily interpreted within the encodingist framework; it seems clear that encodingist intuitions have funded our modeling of representation. Avoiding encodingism through interactivism, however, does allow for the possibility of representation emergence and change. By separating representational content from the aspects of a system that constitute epistemic contact with the world, and providing an account of how they are functionally associated, allows for new representations to be created and existing ones changed by the system itself. And this, in turn, has architectural entailments for how systems which represent will be built and operate, as I have gone on to show.

Fourth, *I have shown how an artificial life model (described in Section 5) exhibits real representational emergence.* The creature is capable of building its own representations of the objects it encounters, and I demonstrated how its representational power can be iteratively increased by increasing the complexity of the creature's interactive capabilities and its environment, and its internal organization-constructing mechanisms. And while the origin of these constructive processes must still be explained, it is clear that any representational content itself that exists in the model does not exist prior to any construction, but does so after such construction — and the creature itself, with its constructive mechanisms, is responsible for the construction of such representation.

And finally, *I have indicated the path from here if we are to implement systems in which representational relational assemblies — situated representation capable of representing the kind of structure in the environment that full-scale analogy-making appears to require — could emerge.* As I have argued in Sections 6 & 7 of Chapter 3, it seems clear that representation emergence and change occurs *prior to* and *as a result of* analogy-making. The mounting pressure of these arguments suggest that we will profit from abandoning encoding-requiring architectures and adopt an approach that allows for emergence and change *in* analogy-making. Such emergence, I contend, is necessary for a full account of *creative* analogy-making — and any other cognitive phenomena that requires the possibility of representation emergence and change. The situated representation framework for modelling representation is the key to such an account.

Bibliography

- Agre, P. E. (1995). Computational research on interaction and agency. *Artificial Intelligence*, 72, 1-52.
- Agre, P. E. & Chapman, D. (1987). Pengi: an implementation of a theory of activity. In *Proceedings for the Sixth National Conference on Artificial Intelligence* (pp.268-272). San Mateo, CA: Morgan Kaufmann.
- Anderson, S. M. & Cole, S. W. (1990). "Do I know you?": The role of significant others in general social perception. *Journal of Personality and Social Psychology*, 59, 384-399.
- Angeline, P. J. (1993). Evolutionary Algorithms and Emergent Intelligence. Unpublished Ph.D. Dissertation. The Ohio State University.
- Armstrong, D. M. (1981). The Causal Theory of the Mind. Reprinted in W. Lycan (ed.). (1991). *Mind and Cognition*. Cambridge: Blackwell Publishers.
- Aronson, J. L. (1984). *A Realist Philosophy of Science*. New York: St. Martin's Press.
- Aronson, J. L., Harré, R. & Way, E. C. (1995). *Realism Rescued: How Scientific Progress Is Possible*. Chicago: Open Court.
- Barsalou, L. W. (1983). Ad hoc categories. *Memory & Cognition*, 11, 211-227
- Barsalou, L. W. (1987). The instability of graded structure: implications for the nature of concepts. In U. Neisser (ed.), *Concepts and Conceptual Development* (pp.101-140, Ch. 5). London: Cambridge University Press.
- Beer, R. D. (1990). *Intelligence as adaptive behavior: An experiment in computational neuroethology*. Cambridge, MA: Academic Press.
- Bickhard, M. H. (1978). The nature of developmental stages. *Human Development*, 21, 217-233.
- Bickhard, M. H. (1980a). A Model of Developmental and Psychological Processes. *Genetic Psychology Monographs*, 102, 61-116.

- Bickhard, M. H. (1980b). *Cognition, Convention, and Communication*. New York: Praeger.
- Bickhard, M. H. (1987). The Social Nature of the Functional Nature of Language. In M. Hickmann (ed.), *Social and Functional Approaches to Language and Thought* (pp.39-65). New York: Academic.
- Bickhard, M. H. (1988). Piaget on Variation and Selection Models: Structuralism, Logical Necessity, and Interactivism. *Human Development*, 31, 274-312.
- Bickhard, M. H. (1992a). How Does the Environment Affect the Person? In L. T. Winegar & J. Valsiner, (eds.), *Children's Development within Social Contexts: Metatheory and Theory* (pp.63-92). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bickhard, M. H. (1992b). Scaffolding and Self Scaffolding: Central Aspects of Development. In L. T. Winegar & J. Valsiner, (eds.), *Children's Development within Social Contexts: Research and Methodology* (pp.33-52). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Bickhard, M. H. (1993). Representational Content in Humans and Machines. *Journal of Experimental and Theoretical Artificial Intelligence*, 5, 285-333.
- Bickhard, M. H. (1995). Intrinsic constraints on language: grammar and hermeneutics. *Journal of Pragmatics*, 23, 541-554.
- Bickhard, M. H. (1998). A process model of the emergence of representation. Presented at the International Conference on Emergence, August 1998.
- Bickhard, M. H. (in press). Levels of Representationality. *Journal of Experimental and Theoretical Artificial Intelligence*.
- Bickhard, M. H. & Campbell, D. T. (1998a). Emergence. Unpublished Manuscript. To appear in P. B. Anderson, N. O. Finnemann, C. Emmeche & P. V. Christiansen (Eds.) *Emergence and Downward Causation*.
- Bickhard, M. H. & Campbell, D. T. (1998b). Variations in variation and selection: the ubiquity of the variation-and-selection-retention ratchet in emergent organizational complexity. Unpublished Manuscript.
- Bickhard, M. H. & Campbell, R. L. (1992). Some foundational questions concerning language studies: with a focus on categorial grammars and model theoretic possible worlds semantics. *Journal of Pragmatics*, 17 (5/6), 104-433.
- Bickhard, M. H. & Campbell, R. L. (1996). Topologies of Learning and Development. *New Ideas in Psychology*, 14 (2), 111-156.

- Bickhard, M. H. & Richie, D. M. (1983). *On The Nature of Representation: A Case Study of James Gibson's Theory of Perception*. New York: Praeger.
- Bickhard, M. H. & Terveen, L. (1995). *Foundational Issues in Artificial Intelligence and Cognitive Science - Impasse and Solution*. New York: Elsevier Scientific.
- Black, M. (1962). *Models and Metaphors*. Ithaca, NY: Cornell University Press.
- Block, N. (1986). Advertisement for a semantics for psychology. In P. A. French, T. E. Uehling & H. K. Wettstein (eds.), *Midwest Studies in Philosophy X: Studies in the Philosophy of Mind* (pp. 615-678). Minnesota.
- Blum, L., Shub, M., and Smale, S. (1989). On a theory of computation and complexity over the real numbers: NP-completeness, recursive functions, and universal turing machines. *Bulletin of the American Mathematical Society*, 21 (1), 1-46.
- Boden, M. A. (1977). *Artificial Intelligence and Natural Man*. New York: Basic Books.
- Boden, M. A. (1991). *The Creative Mind: Myths and Mechanisms*. New York: Basic Books.
- Braitenberg, V. (1984). *Vehicles*. Cambridge, MA: The MIT Press
- Brooks, R. A. (1989). A robot that walks: emergent behavior from a carefully evolved network. *Neural Computation*, 1 (2), 253-262.
- Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47, 139-159.
- Brown, A. L. (1989). Analogical learning and transfer: What develops? In S. Vosniadou & A. Ortony (eds), *Similarity and analogical reasoning* (pp. 369-412). New York: Cambridge University Press.
- Brown, A. L. (1990). Domain specific principles affect learning and transfer in children. *Cognitive Science*, 14, 107-134.
- Brown, A. L. & Kane, M. J. (1988). Preschool children can learn to transfer: Learning to learn and learning from example. *Cognitive Psychology*, 20, 493-523.
- Bruner, J. S. (1957). Going Beyond the Information Given. Reprinted in J. S. Bruner & J. M. Anglin (eds.). (1973). *Beyond The Information Given: Studies in the Psychology of Knowing*. New York: W. W. Norton & Company.
- Burstein, M. H. (1988). Incremental learning from multiple analogies. In A. Prieditis (ed.), *Analogica* (pp.37-62). Los Altos, CA: Morgan Kaufmann.

- Camac, M. K. & Glucksberg, S. (1984). Metaphors do not use associations between concepts, they are used to create them. *Journal of Psycholinguistic Research*, 13, 443-455.
- Campbell, N. R. (1957). *Foundations of Science*. New York: Dover.
- Campbell, R. L. (1992). Clearing the ground: foundational questions once again. *Journal of Pragmatics*, 17 (5/6), 557-602.
- Campbell, R. L. & Bickhard, M. H. (1986). *Knowing Levels and Developmental Stages*. Basel: Karger.
- Chalmers, D. J. (1994). A Computational Foundation for the Study of Cognition. Manuscript. (Exerpts from manuscript found in *Minds & Machines*, 4 (5), 1-26.)
- Chalmers, D. J. (1996). *The Conscious Mind: In Search of a Fundamental Theory*. New York: Oxford University Press.
- Chalmers, D. J., French, R. M. & Hofstadter D. R. (1992). High-level perception, representation, and analogy: A critique of artificial intelligence methodology. *Journal of Experimental & Theoretical Artificial Intelligence* 4 (3), 185-211.
- Clancey, W. J. (1992). The frame of reference problem in the design of intelligent machines. In K. VanLehn (ed.) *Architectures for Intelligence: The Twenty-Second Carnegie Symposium on Cognition* (pp. 357-423). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Clancey, W. J. (1997). *Situated Cognition: On Human Knowledge and Computer Representations*. New York: Cambridge University Press.
- Clark, A. (1989). *Microcognition: Philosophy, Cognitive Science, and Parallel Distributed Processing*. Cambridge, MA: The MIT Press.
- Clark, A. (1993). *Associative Engines: Connectionism, Concepts, and Representational Change*. Cambridge, MA: The MIT Press.
- Davidson, D. (1970). Mental events. In L. Foster & J. Swanson (eds.), *Experience and Theory*. London: Duckworth.
- Dennett, D. (1969). *Content and Consciousness*. London: Routledge & Kegan Paul.
- Defays, D. (1986). *Numbo: A study in cognition and recognition*. CRCC Report No. 13. Center for Research on Concepts and Cognition, Indiana University, Bloomington, Ind. Also in Hofstadter, D. R. (1995), *Fluid Concepts and Creative Analogies*. New York: Basic Books.

- Dretske, F. (1981). *Knowledge and the Flow of Information*. Cambridge, MA: The MIT Press.
- Dietrich, E. (1985). Computer Thought: Propositional attitudes and metaknowledge. PhD Dissertation, Dept. of Philosophy, University of Arizona.
- Dietrich, E. (1990). Computationalism. *Social Epistemology*, 4 (2), 135-154. And reprinted in, E. Dietrich (ed.) (1994), *Thinking Computers and Virtual Persons*. Academic Press, Inc.
- Dietrich, E. (1994). Thinking Computers and The Problem of Intentionality. In E. Dietrich (ed.), *Thinking Computers and Virtual Persons*. Academic Press, Inc.
- Dietrich, E. (1995). Book Review: Dorothy L. Cheney and Robert M. Seyfarth, *How Monkeys See the World*. *Minds and Machines*, 5, 1-8.
- Dietrich, E. (1996). The role of the Frame Problem in Fodor's Modularity Thesis. In K. Ford & Z. Pylyshyn (eds.), *The Robots Dilemma Revisited*. (pp.9-24). Cambridge, MA: The MIT Press.
- Dietrich, E. (in press). Analogical reminding and conceptual change, or you can't step into the same mind twice. In E. Dietrich & A. B. Markman (eds.), *Cognitive Dynamics*. Cambridge, MA: The MIT Press.
- Dietrich, E. & Beyer, D. (1998). DogPaste: an excursion in situated analogy. Unpublished Manuscript.
- Dietrich, E. & Markman, A. D. (in press). *Cognitive Dynamics*. Cambridge, MA: The MIT Press.
- Dietrich, E., Morrison, C. & Oshima, M. (1996). Conceptual change as change of inner perspective. In *Papers from the 1996 AAAI Fall Symposium*. (November 9-11, Cambridge, MA). Technical Report FS-96-02. Menlo Park, CA: AAAI Press, pp.37-41.
- Drescher, G. L. (1991). *Made-up minds: A constructivist approach to artificial intelligence*. Cambridge, MA: The MIT Press.
- Falkenhainer, B. (1987). An examination of the third stage in the analogy process: Verification-based analogical learning. *Proceedings of IJCAI-87* (pp. 260-263).
- Falkenhainer, B. (1988). Learning from physical analogies: a study in analogy and the explanation process. PhD thesis, University of Illinois at Urbana-Champaign.

- Falkenhainer, B. (1990a). A unified approach to explanation and theory formation. In Shrager & Langley (eds.), *Computational Models for Scientific Discovery and Theory Formation* (San Mateo, CA: Morgan Kaufmann). Also in Shavlik and Dietterich (Eds.), *Readings in Machine Learning* (San Mateo, CA: Morgan Kaufmann).
- Falkenhainer, B. (1990b). Analogical interpretation in context. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp.69-76). Hillsdale, NJ: Lawrence Erlbaum Associates
- Falkenhainer, B., Forbus, K. D. & Gentner, D. (1986). The structure-mapping engine. In *Proceedings of the fifth national conference on artificial intelligence* (pp.272-277). Los Altos, CA: Morgan Kaufmann.
- Falkenhainer, B., Forbus, K. D. & Gentner, D. (1989). The structure-mapping engine: algorithm and examples. *Artificial Intelligence*, 41 (1), 1-63.
- Ferguson, R. (1994). MAGI: Analogy-based encoding using regularity and symmetry. In A. Ram & K Eiselt (eds.), *Proceedings of the sixteenth annual conference of the Cognitive Science Society* (pp.283-288). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Ferguson, R. W., Aminoff, A. & Gentner, D. (1996). Modeling qualitative differences in symmetry judgments. In *Proceedings of the Eighteenth Annual Conference of the Cognitive Science Society* (pp. 534-539). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Fodor, J. A. (1974). Special Sciences, or the Disunity of Science as a Working Hypothesis. *Synthese*, 28, 97-115.
- Fodor, J. A. (1987). *Psychosemantics*. Cambridge, MA: The MIT Press
- Fodor, J. A. (1990). *A Theory of Content*. Cambridge, MA: The MIT Press.
- Fodor, J. A. (1994). *The Elm and the Expert*. Cambridge, MA: The MIT Press.
- Forbus, K. D. (1984). Qualitative process theory. *Artificial Intelligence*, 24, 5-168.
- Forbus, K. D., Ferguson, R. W. & Gentner, D. (1994). Incremental structure-mapping. In A. Ram & K Eiselt (eds.), *Proceedings of the sixteenth annual conference of the Cognitive Science Society* (pp.313-318). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Forbus, K. D., Gentner, D. & Law, K. (1995). MAC/FAC: A model of similarity-based retrieval. *Cognitive Science*, 19, 141-205.

- Forbus, K. D., Gentner, D., Markman, A. B. & Ferguson, R. W. (*in press*). Analogy just looks like high level perception: Why a domain-general approach to analogical mapping is right. *Journal of Experimental and Theoretical Artificial Intelligence*.
- French, R. M. (1995). *The subtlety of sameness: A theory and computer model of analogy-making*. Cambridge, MA: The MIT Press
- French, R. M. (1997). When coffee cups are like old elephants — or — Why representation modules don't make sense. In *Proceedings of the 1997 International Conference on New Trends in Cognitive Science* (pp. 158-163), A. Riegler & M. Peschl (Eds.), Austrian Society for Cognitive Science.
- Gentner, D. (1977a). Children's performance on a spatial analogies task. *Child Development*, 48, 1034-1039
- Gentner, D. (1977b). If a tree had a knee, where would it be? Children's performance on simple spatial metaphors. *Papers and Reports on Child Language Development*, 13, 157-164.
- Gentner, D. (1983). Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7, 155-170.
- Gentner, D. (1988). Analogical inference and analogical access. In A. Prieditis (ed.), *Analogica* (pp.63-88). Los Altos, CA: Morgan Kaufmann.
- Gentner, D. (1989). The mechanisms of analogical learning. In S. Vosniadou & A. Ortony (eds.), *Similarity and analogical reasoning* (pp.199-241). London: Cambridge University Press.
- Gentner, D. & Clement, C. (1988). Evidence for relational selectivity in the interpretation of analogy and metaphor. In G. H. Bower (ed.), *The psychology of learning and motivation* (Vol. 22, pp.307-358). New York: Academic Press.
- Gentner, D., Falkenhainer, B. & Skorstad, J. (1987, January). Metaphor: The good, the bad, and the ugly. In *Proceedings of the Third Conference on Theoretical Issues in Natural Language Processing* (pp.155-159), Las Cruces, NM.
- Gentner, D. & Forbus, K. D. (1991). MAC/FAC: A model of similarity-based retrieval. In *Proceedings of the Thirteenth Annual Cognitive Science Society*. Hillsdale, NJ: Lawrence Erlbaum Associates.
- Gentner, D. & Landers, R. (1985). Analogical reminding: A good match is hard to find. In *Proceedings International Conference on Systems, Man and Cybernetics*, Tucson, AZ.

- Gentner, D. & Markman, A. B. (1995). Analogy is like similarity. In C. Cacciari (ed.), *Similarity*. Brussels: Bropels.
- Gentner, D. & Markman, A. B. (1997). Structure Mapping in Analogy and Similarity. *American Psychologist*, 52 (1), 45-56.
- Gentner, D. & Rattermann, M. J. (1991). Language and the career of similarity. In S. A. Gelman and J. P. Byrnes, (eds.), *Perspectives on language and thought interrelations in development* (pp.225-277). London: Cambridge University Press.
- Gentner, D., Rattermann, M. J., Markman, A. B., & Kotovsky, L. (1995). Two Forces in the Development of Relational Similarity. In T. J. Simon & G. S. Halford (eds.), *Developing Cognitive Competence: New Approaches to Process Modelling* (pp.263-313). Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.
- Gentner, D. & Toupin, C. (1986). Systematicity and surface similarity in the development of analogy. *Cognitive Science*, 10, 277-300.
- Gentner, D., & Wolff, P. (*in press*). Metaphor and knowledge change. In E. Dietrich & A. Markman (Eds.), *Cognitive dynamics: Conceptual change in humans and machines*. Cambridge, MA: The MIT Press.
- Gick, M. L. & Holyoak, K. J. (1983). Schema induction and analogical transfer. *Cognitive Psychology*, 15, 1-38.
- Giere, R. N. (1988). *Explaining Science: A Cognitive Approach*. Chicago: The University of Chicago Press.
- Giere, R. N. (1997). *Understanding Scientific Reasoning* (4th Ed.). Chicago: Holt, Rinehart and Winston, Inc.
- Gibson, J. J. (1950). *The Perception of the Visual World*. Boston, MA: Houghton Mifflin.
- Gibson, J. J. (1966). *The Senses Considered as Perceptual Systems*. Boston, MA: Houghton Mifflin.
- Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Boston, MA: Houghton Mifflin.
- Gleick, J. (1987). *Chaos: Making a New Science*. New York: Penguin.
- Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley Publishing Company, Inc.

- Goldstone, R. L. (1994). Similarity, interactive activation, and mapping. *Journal of Experimental Psychology: Learning, Memory and Cognition*, 20, 3-28.
- Goldstone, R. L. & Medin, D. L. (1994). The time course of comparison. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 20, 29-50.
- Goldstone, R. L., Schyns, P. G. & Medin, D. L. (1997). Learning to bridge between perception and cognition. In R. L. Goldstone, P. G. Schyns & D. L. Medin (eds.), *The Psychology of Learning and Motivation* (Vol. 36, pp.1-14). San Diego, CA: Academic Press.
- Goldstone, R. L., Steyvers, M., Spencer-Smith, J. & Kersten, A. (in press). Interactions between perceptual and conceptual learning. In E. Dietrich & A. B. Markman (eds.), *Cognitive Dynamics*. Cambridge, MA: The MIT Press.
- Goodman, N. (1972). *Problems and prospects*. Indianapolis, IN: Bobbs-Merrill.
- Goswami, U. (1992). *Analogical Reasoning In Children*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.
- Halford, G. S. (1992). Analogical reasoning and conceptual complexity in cognitive development. *Human Development*, 35 (4), 193-217.
- Halford, G. S. (1993). *Children's Understanding: The Development of Mental Models*. Hillsdale, NJ: Lawrence Erlbaum Associates, Publishers.
- Hall, R. P. (1988). Computational approaches to analogical reasoning: A comparative analysis. *Artificial Intelligence*, 39, 39-120.
- Hanson, N. R. (1958). *Patterns of Discovery*. Cambridge University Press.
- Hanson, N. R. (1963). *The Concept of the Positron*. Cambridge University Press.
- Harnad, S. (ed.) (1987). *Categorical Perception: The Groundwork of Cognition*. New York: Cambridge.
- Harnad, S. (1990). The symbol grounding problem. *Physica D*, 42, 335-346.
- Harré, R. (1970). *The Principles of Scientific Thinking*. Chicago: The University of Chicago Press.
- Harré, R. (1983). *An Introduction to the Logic of the Sciences*. (2nd Ed.). New York: St. Martin's Press.
- Harré, R. (1986). *Varieties of Realism*. New York: Basil Blackwell Inc.

- Hayes, P. & Ford, K. (1995). Turing Test considered harmful. In *Proceedings of IJCAI95* (pp.972-977). Cambridge, MA: The MIT Press.
- Hendriks-Jansen, H. (1994). In praise of interactive emergence, or why explanations don't have to wait for implementations. In R. A. Brooks & P. Maes (eds.), *Artificial Life IV* (Proceedings of the Fourth International Workshop on the Synthesis and Simulations of Living Systems), (pp.70-79). Cambridge, MA: The MIT Press.
- Hendriks-Jansen, H. (1996). *Catching Ourselves in the Act: Situated Activity, Interactive Emergence, Evolution, and Human Thought*. Cambridge, MA: The MIT Press.
- Hesse, M. B. (1966). *Models and Analogies in Science*. University of Notre Dame Press.
- Hesse, M. B. (1974). *The Structure of Scientific Inference*. Berkeley: University of California Press.
- Hoffman, R. R. (1995). Monster Analogies. *AI Magazine, AAAI*, Fall 1995, 11-35.
- Hofstadter, D. R. (1979). *Gödel, Escher, Bach: An eternal golden braid*. New York: Basic Books.
- Hofstadter, D. R. (1981). Metamagical Themas. *Scientific American*, 245 (3), 409-415.
- Hofstadter, D. R. (1983). The architecture of Jumbo. *Proceedings of the International Machine Learning Workshop*. Monticello, IL.
- Hofstadter, D. R. (1984). The Copycat project: An experiment in nondeterminism and creative analogies. AI Memo No. 755, Massachusetts Institute of Technology. Cambridge, MA.
- Hofstadter, D. R. (1985). Analogies and roles in human and machine thinking. In Hofstadter, D. R. (ed.), *Metamagical themas* (pp.547-603). New York: Basic Books.
- Hofstadter, D. R. & the Fluid Analogies Research Group (FARG). (1995). *Fluid Concepts and Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*. New York: BasicBooks, A Division of Harper Collins Publishers, Inc.
- Hofstadter, D. R., Clossman, G. & Meredith, M. J. (1980). *Shakespeare's plays weren't written by him, but by someone else of the same name. An essay on intensionality and frame-based knowledge representation systems*. Technical Report No. 96. Computer Science Department, Indiana University, Bloomington, Ind.

- Hofstadter, D. R. & Mitchell, M. (1991). The Copycat project: A model of mental fluidity and analogy-making. CRCC Technical Report 58. Center for Research on Concepts and Cognition, Indiana University, Bloomington, IN. Reprinted in J. Barnden & K. Holyoak (eds.), *Advances in connectionist and neural computation theory*. Vol. 2: *Analogical connections*.
- Holland, J. H. (1992). *Adaptation In Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence* (2nd edition). Cambridge: The MIT Press. (First Edition published in 1975.)
- Holyoak, K. J. & Thagard, P. (1989). Analogical mapping by constraint satisfaction. *Cognitive Science*, 13 (3), 295-355.
- Holyoak, K. J. & Thagard, P. (1995). *Mental Leaps: Analogy in Creative Thought*. Cambridge, MA: Bradford.
- Inhelder, B. & Piaget, J. (1958). *The growth of logical thinking from childhood to adolescence*. New York: Basic Books.
- Juster, Norton. (Illustrations by Jules Feiffer). (1988). *The Phantom Tollbooth*. New York: Bullseye Books. (Originally published in 1961; New York: Epstein & Carroll - distributed by Random House.)
- Kahneman, D. & Miller, D. T. (1986). Norm theory: Comparing reality to its alternatives. *Psychological Review*, 93, 136-153.
- Karmiloff-Smith, A. (1992). *Beyond Modularity: A Developmental Perspective on Cognitive Science*. Cambridge, MA: The MIT Press.
- Keane, M. T., Ledgeway, T. & Duff, S. (1994). Constraints on analogical mapping: A comparison of three models. *Cognitive Science*, 18, 387-438.
- Kim, J. (1978). Supervenience and nomological incommensurables. *American Philosophical Quarterly*, 15, 149-156.
- Kim, J. (1993). *Supervenience and Mind*. Cambridge, MA: Cambridge University Press.
- Kirkpatrick, S., Gelatt Jr., C. D. & Vecchi, M. P. (1983). Optimization by simulated annealing. *Science*, 220, 671-680.
- Kirsh, D. (1991). Today the earwig, tomorrow man? *Artificial Intelligence*, 47, 161-184.

- Kolstad, V. & Baillargeon, R. (1991). *Appearance and knowledge-based responses to containers in infants*. Unpublished manuscript.
- Kotovskiy, L. & Gentner, D. (1990). Pack Light: You Will Go Farther. In *Proceedings of the Second Midwest Artificial Intelligence and Cognitive Science Society Conference*. (pp.60-72). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Kotovskiy, L. & Gentner, D. (1994). Progressive alignment: A mechanism for the development of relational similarity. Unpublished Manuscript.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge: The MIT Press.
- Kuhn, T. (1970). *The Structure of Scientific Revolutions* (2nd Ed.). Chicago: The University of Chicago Press.
- Langley, P., Simon, H. A., Bradshaw, G. L. & Zytkow, J. M. (1987). *Scientific Discovery: Computational Explorations of the Creative Process*. Cambridge, MA: The MIT Press.
- Lenat, D. B. (1976). AM: An artificial intelligence approach to discovery in mathematics in heuristic search. Ph.D. dissertation. Computer Science Department, Stanford University. Also available as Stanford University Computer Science Department technical report STAN-CS-76-570 and Stanford University Artificial Intelligence Laboratory memo AIM-286.
- Lenat, D. B. (1979). On automated scientific theory formation: a case study using the AM program. In J. E. Hayes, D. Michie, and O. I. Mikulich (eds.), *Machine Intelligence 9* (pp.251-283.). Ellis Horwood, Chister.
- Lenat, D. B. (1982). AM: discovery in mathematics as heuristic search. In R. Davis and D. Lenat (eds.), *Knowledge-Based Systems in Artificial Intelligence* (pp.1-25). New York: McGraw-Hill
- Lenat, D. B. (1983). EURISKO: A program that learns new heuristics and domain Concepts. *Artificial Intelligence*, 21 (1, 2), 61-98.
- Lenat, D. B. & Brown, J. S. (1984). Why AM and EURISKO appear to work. *Artificial Intelligence*, 23, 269-294.
- Lewis, D. (1983). *Philosophical Papers, Vol. 1*. Oxford University Press.
- Lindgren, K. & Norahl, M. G. (1994). Cooperation and community structure in artificial ecosystems. *Artificial Life*, 1, 15-37.

- Lloyd, G. E. R. (1966). *Polarity and Analogy: Two Types of Argumentation in Early Greek Thought*. Cambridge, U.K.: Cambridge University Press.
- Loren, L., Dietrich, E., Morrison, C. T. & Beskin, J. (in press). What it means to be situated. *Cybernetics and Systems*.
- Lovelace, A. Notes on Manabrea's Sketch of the Analytical Engine Invented by Charles Babbage. In B. V. Bowden (ed.), *Faster Than Thought* (London, 1953), p.362-408.
- Maier, N. R. F. (1931). Reasoning in humans: II. The solution of a problem and its appearance in consciousness. *Journal of Comparative Psychology*, 12, 181-194.
- Marcus, G. (in press). "Two kinds of representation" in E. Dietrich and A. Markman (eds.), *Cognitive Dynamics*, MIT.
- Markman, A. B. & Dietrich, E. (1998). "In defense of representation" Unpublished Manuscript.
- Markman, A. B. & Gentner, D. (1993). Splitting the differences: a structural alignment view of similarity. *Journal of Memory and Language*, 32, 431-467.
- Mataric, M. J. (1992). Integration of representation into goal-driven behavior-based robots. *IEEE Transactions on Robotics and Automation*, 8 (3), 304-312.
- Mataric, M. J. & Brooks, R. A. (1990). Learning a distributed map representation based on navigation behaviors. In *Proceedings of 1990 USA Japan Symposium on Flexible Automation* (pp.499-506). Kyoto, Japan.
- McClamrock, R. (1995). *Existential Cognition*. Chicago: The University of Chicago Press.
- McDermott, D. (1976). Artificial intelligence meets natural stupidity. *SIGART Newsletter*, no. 57, April 1976. Reprinted in J. Haugeland (ed.), *Mind Design: Philosophy, Psychology, Artificial Intelligence*. (pp.143-160). Montgomery, VT: Bradford Books, 1981.
- Meredith, M. J. (1986). Seek-Whence: a model of pattern perception. Technical Report No. 214. Computer Science Department, Indiana University, Bloomington, Ind.
- Medin, D. L., Goldstone, R. L. & Gentner, D. (1993). Respects for similarity. *Psychological Review*, 100, 254-278.
- Meyer, D. E. & Schvaneveldt, R. W. (1971). Facilitation in recognizing pairs of words: evidence of a dependence between retrieval operations. *Journal of Experimental Psychology*, 90, 227-235.

- Millikan, R. (1984). *Language, Thought and Other Biological Categories: New Foundations for Realism*. Cambridge, MA: The MIT Press
- Millikan, R. (1993). *White Queen Psychology and Other Essays for Alice*. Cambridge, MA: The MIT Press.
- Mitchell, M. (1993). *Analogy-Making as Perception: A Computer Model*. Cambridge, MA: The MIT Press.
- Morrison, C. T. (1997). Are we wrong about representation? *Journal of Experimental and Theoretical Artificial Intelligence*, 9, 441-469.
- Morrison, C. T. & Dietrich E. (1995). Structure-Mapping vs. High-level Perception: the mistaken fight over the explanation of analogy,” In the *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society* (pp.678-682).
- Morrison, C. T. & Lee, C. (1998). The development of representation for analogy-making: the *what* and *when*, but not the *how* of SME. Unpublished Manuscript.
- Nagel, E. (1961). *The Structure of Science: Problems in the Logic of Scientific Explanation*. New York: Harcourt, Brace & World, Inc.
- Oshima, M. (1996). Analogy as a creative process. Master’s thesis, SUNY Binghamton, Binghamton, NY.
- Piaget, J. (1954). *The Child’s Construction of Reality*. New York: Basic Books.
- Piaget, J., Montangero, J. & Billeter, J. (1977). La formation des correlats [The formation of correlations]. In J. Piaget (ed.), *L’Abstraction reflechissante* (pp. 115-129). Paris: Presses Universitaires de France.
- Pinker, S. & Prince, A. (1988). On language and connectionism: analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28, 73-193.
- Plunkett, K. & Marchman, V. (1991). U-shaped learning and frequency effects in a multi-layered perception: implications for child language acquisition. *Cognition*, 38, 43-102.
- Putnam, H. (1967). The nature of mental states. Reprinted in W. Lycan (Ed.). (1991). *Mind and Cognition*. Cambridge: Blackwell Publishers. (p.47).
- Qin, Y. & Simon, H. A. (1990). Laboratory replication of scientific discovery processes. *Cognitive Science*, 14, 281-310.
- Quillian, M. R. (1968). Semantic memory. In *Semantic information processing*, ed. M. Minsky. Cambridge, MA: MIT Press.

- Rattermann, M. J. & Gentner, D. (1990). The development of similarity use: It's what you know, not how you know it. In *Proceedings of the Second Midwest Artificial Intelligence and Cognitive Science Society Conference* (pp. 54-59). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Rattermann, M. J., Gentner, D. & DeLoache, J. (1990). Effects of labels on children's use of relational similarity. In *Proceedings of the Twelfth Annual Conference of the Cognitive Science Society* (pp. 22-29). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Ray, T. S. (1994). An evolutionary approach to synthetic biology: zen and the art of creating life. *Artificial Life, 1*, 179-209.
- Read, S. J. (1984). Analogical reasoning in social judgment: The importance of causal theories. *Journal of Personality and Social Psychology, 46*, 14-25.
- Ritchie, G. & Hanna, F. (1990). AM: a case-study in AI methodology. In D. Partridge and Y. Wilks (eds.), *The Foundations of AI: A Sourcebook*. New York: Cambridge University Press.
- Root-Bernstein, R. S. (1983). Mendel and Methodology. *History of Science, 21*, 275-295.
- Rumelhart, D. E. & McClelland, J. L. (1986). On learning the past tenses of English verbs. In J. McClelland *et al.* (ed.), *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*. Vol. 2 (pp.216-271). Cambridge: The MIT Press.
- Searle, J. (1980). Minds, Brains, and Programs. Reprinted in M. Boden (1990) *The Philosophy of Artificial Intelligence*. Oxford: Oxford University Press.
- Searle, J. (1992). *The Rediscovery of the Mind*. Cambridge, MA: The MIT Press.
- Schacter, D. L. (1987). Implicit memory: history and current status. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 13*, 501-518.
- Schyns, P. G., Goldstone, R. L. & Thibaut, J.-P. (in press). The development of features in object concepts. *Behavioral and Brain Sciences*.
- Simon, H. A. (1989). The scientist as problem solver. In D. Klahr and K. Kotovsky (eds.), *Complex Information Processing*. Hillsdale, NJ: Lawrence Erlbaum.
- Sims, K. (1995). Evolving 3D morphology and behavior by competition. *Artificial Life, 1*, 353-372.

- Smith, E. E. & Medin, D. L. (1981). *Categories and concepts*. Cambridge, MA: Harvard University Press.
- Smolensky, P. (1988). On the proper treatment of connectionism. *Behavioral and Brain Sciences*, *11*, 1-74.
- Suchman, L. A. (1987). *Plans and Situated Actions*. Cambridge, MA: Cambridge University Press.
- Terzopoulos, D., Tu, X. & Grzeszczuk, R. (1995). Artificial fishes: autonomous locomotion, perception, behavior, and learning in a simulated physical world. *Artificial Life*, *1*, 327-351.
- Thelen, E. & Smith, L. B. (1994). *A Dynamic Systems Approach to the Development of Cognition and Action*. Cambridge, MA: The MIT Press.
- Tsotsos, J. K. (1995). Behaviorist intelligence and the scaling problem. *Artificial Intelligence*, *75*, 135-160.
- Turing, A. (1936). On computable numbers with an application to the entscheidungs problem. In *Proceedings of the London Mathematical Society*, series 2, (42), 230-263, (43), 544-546.
- Tversky, A. (1977). Features of similarity. *Psychological Review*, *84*, 327-352.
- Tversky, A. & Kahneman, D. (1974). Judgment under uncertainty: heuristics and biases. *Science*, *185*, 1124-1131.
- van Gelder, T. J. & Port, R. (1995). *Mind as Motion: Explorations in the Dynamics of Cognition*. Cambridge, MA: The MIT Press.
- Vosniadou, S. & Ortony, A. (1989). Similarity and analogical reasoning: a synthesis. In S. Vosniadou & A. Ortony (eds.), *Similarity and analogical reasoning* (pp.199-241). London: Cambridge University Press.
- Weizenbaum, J. (1966). ELIZA — a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9* (1), 36-44.
- Weizenbaum, J. (1976). *Computer Power and Human Reason: From Judgment to Calculation*. San Francisco: Freeman.
- Wheeler, M. (1996). The Philosophy of Situated Activity. Ph.D. Dissertation. The University of Sussex.
- Wilson, P. (1991). Pointer swizzling at page fault time: efficiently supporting huge address spaces on standard hardware. *Computer Architecture News*, *19*, 6-13.

Wittgenstein, L. (1953). *Philosophical Investigations*. Oxford: Blackwell.

Clayton Thomas Morrison

Vita

- Date of Birth:** 12 March 1970
- Place of Birth:** Sierra Madre, California
- Education:** Ph.D. in Philosophy, expected August 1998
Binghamton University (SUNY), Binghamton, NY
- M.A. in Philosophy, January 1995
Binghamton University (SUNY), Binghamton, NY
- B. A., June 1992
Occidental College, Los Angeles, CA
- Honors/Awards:** Dissertation Year Fellowship, Binghamton
University, 1997-1998
- Graduate Student Award for Excellence
in Teaching, Binghamton University, 1997
- Ringle Award for Academic Excellence,
Binghamton University, 1995
- Teaching Assistantship, Binghamton University,
1993-1997
- Distinguished Honors for Senior Comprehensives
Project, Occidental College, 1992
- Ford Fellowship, Occidental College, Summer 1991
- Scholastic and
Professional Experience:** Adjunct Lecturer, Binghamton University,
Spring 1995, Summers of 1995, 1996, 1997
- Teaching Assistant, Binghamton University,
1993-1997

Teacher Intern, Skills Enrichment Program.
The Chandler School (Johns Hopkins
University, Center for Talented Youth),
Summer 1994

Publications:

- Lee, C., van Heuveln, B. , Morrison, C. T. & Dietrich, E. (1998). "Why Connectionist Nets Are Good Models — Commentary on Green on Connectionist-Explanation." *Psychology* 9(17)
([ftp://ftp.princeton.edu/pub/harnad/Psycology/1998.volume.9/psycology.98.9.17.connectionist explanation.14.green](ftp://ftp.princeton.edu/pub/harnad/Psycology/1998.volume.9/psycology.98.9.17.connectionist%20explanation.14.green))
- Loren, L., Dietrich, E., Morrison, C. T. & Beskin, J. (in press). "What it means to be situated." *Cybernetics and Systems*.
- Morrison, C. T. (1997). "Essay Review: Are we wrong about representation? A review of Bickhard & Terveen (1995)." *Journal of Experimental and Theoretical Artificial Intelligence*, 9, 441-469.
- Dietrich, E., Morrison, C. T. & Oshima, M. (1996). "Conceptual Change as Change of Inner Perspective." In *Papers from the 1996 AAAI Fall Symposium*. (November 9-11, Cambridge, MA). Technical Report FS-96-02. Menlo Park, CA: AAAI Press. pp.37-41.
- Morrison, C. T. & Dietrich, E. (1995). "Structure-Mapping vs. High-level Perception: the mistaken fight over the explanation of analogy," in the *Proceedings of the Seventeenth Annual Conference of the Cognitive Science Society*, pp.678-682.

Professional Societies:

American Association for Artificial Intelligence
American Philosophical Association
Cognitive Science Society
Society for Machines and Mentality
Society for Philosophy and Psychology

Contact Information:

Binghamton University
Department of Philosophy
PACCS Program
Hinman 129
Binghamton, NY 13902-6000

clayton@turing.paccs.binghamton.edu
<http://www.paccs.binghamton.edu/clayton>