Curious Design Agents and Artificial Creativity

A Synthetic Approach to the Study of Creative Behaviour

Rob Saunders
B.Sc. (Hons.)

A thesis submitted in fulfilment of the requirements for the degree of Doctor of Philosophy

February 2002

Department of Architectural and Design Science
Faculty of Architecture
University of Sydney

In loving memory of my father.

Acknowledgements

According to Schulman and Cox (1997) this is the only part of my thesis that I can be sure anyone will ever look at – so I guess I had better make it a good read.

First and foremost, I wish to thank my supervisor John Gero for his insights, guidance and support throughout the course of my research. I'd like to thank the academic staff of the Faculty of Architecture including Terry Purcell, Glenn Hill, Mike Rosenman and Scott Chase for their encouragement and helpful discussions. I'd also like to thank my fellow students in the Key Centre of Design Computing and Cognition for providing a stimulating research environment, in particular I would like to thank: Michael Tang, Julie Jupp, Ricardo Sosa, Alex Summerfield, Justin Clayden, Vasilije Kokotovich, Thorsten Schnier, Udo Kannengiesser, Stephen Clarke, Jarek Kulinsky, and Rabee Reffat.

Thanks go to my "muses" Sara Broomhall, James Rutherford, Marieke Moolenaar, Chris Jenkins, and Gary Martin for their inspiration: without your brilliant insights my thesis would be much the poorer. Thanks also to my many wonderful friends in Australia, the UK and other parts of the world that have seen me through good times and bad; especially Ruby Lin, Paul Martin, Ingrid Evans, Nikki Farrar, Rose Bradshaw, Hiroshi Narushima, Doug Falconer, Tracey Hall, Bruce Lamond, Dave Dickerson, Marcus Alphey, Graham Clark, Steven Webster, Isabelle Watson, and Silke Machate.

Finally, this thesis would not exist if it weren't for my family. I cannot thank you enough for all the love and support that a son and brother could ever hope for. In particular I thank my mother and father, for giving me an unquenchable thirst for knowledge as a child and helping me to find the courage to go wherever it takes me.

Abstract

Creative products are generally recognised as satisfying two requirements: firstly they are useful, and secondly they are novel. Much effort in AI and design computing has been put into developing systems that can recognise the usefulness of the products that they generate. In contrast, the work presented in this thesis has concentrated on developing computational systems that are able to recognise the novelty of their work. The research has shown that when computational systems are given the ability to recognise both the novelty and the usefulness of their products they gain a level of autonomy that opens up new possibilities for the study of creative behaviour in single agents and the emergence of social creativity in multi-agent systems.

The work presented in this thesis has developed a model of curiosity in design as the selection of design actions with the goal of generating novel artefacts. Agents that embody this model of curiosity are called "curious design agents". The behaviour of curious design agents is demonstrated with a range of applications to visual and non-visual design domains. Visual domains include rectilinear drawings, Spirograph patterns, and "genetic artworks" similar to the work of Karl Sims. Non-visual domains include an illustrative abstract design space useful for visualising the behaviour of curious agents and the design of doorways to accommodate the passage of large crowds. The design methods used in the different domains show that the model of curiosity is applicable to models of designing by direct manipulation, parametric configuration or by using a separate design tool that embodies the generative aspects of the design process.

In addition, an approach to developing multi-agent systems with autonomous notions of creativity called artificial creativity is presented. The opportunities for studying social creativity in design are illustrated with an artificial creativity system used to study the emergence of social notions of whom and what are creative in a society of curious design agents. Developing similar artificial creativity systems promises to be a useful synthetic approach to the study of socially situated, creative design.

Table of Contents

Chapter 1

Introduction

This thesis is concerned with the computational modelling of creativity. Consequently, it is also concerned with one of the oldest questions raised about computers: Can computers be creative? Objections to the notion that computers could ever be creative pre-date by over a century the invention of the first practical computers with which to investigate the question empirically. Famously, Ada Augusta, Countess of Lovelace, commented upon her translation of Menabrea's "Sketch of the Analytical Engine", declaring that: "The Analytical Engine has no pretensions whatever to *originate* anything. It can do [only] *whatever we know how to order it* to perform" (emphasis added by Boden, 1990). In Lady Lovelace's opinion, any creative products of the Analytical Engine would have to be credited not to the engine, but to the engineer.

Turing recognised the importance of creativity in any definition of intelligence when he attempted to answer Lovelace's objection in his seminal paper "Computer Machinery and Intelligence", the same paper in which he introduced his now famous test for machine intelligence (Turing, 1950). Turing suggested that objections to the possibility of computers being creative of the type put forward by Lady Lovelace were based on a common misunderstanding of the nature of reasoning in the mind, resulting in an over-statement of the powers of rational thought. In particular, Turing pointed out that a person knowing a set of facts and rules about the world does not mean that the person immediately knows all of the implications of applying the rules to the facts.

Turing suggested that a better variant of Lovelace's objection would be that a machine can never 'take us by surprise' but he then proceeded to declare that computers often surprised him because of his own faulty understanding of what he had 'order them to perform'. In making this argument Turing tried to show that the engineer would be no more responsible for the creativity of a machine than the machine itself because the engineer could not predict the creative behaviour at design time. Turing's argument does not provide us with much information about the possible processes involved in creative thinking but it does highlight the importance of emergence, novelty, and surprise in computational models of creativity.

To be able to study machine creativity within the scope of a single thesis, it is necessary to restrict the discussion to address more limited questions than whether computers can be creative. Following the lead of Turing, and more recently Boden (1990), two questions are asked here. Firstly, can computers model being 'surprised'? Secondly, can the novelty of a surprising discovery motivate the production of creative works?

1.1 MOTIVATIONS

The initial motivation for this research has come from existing computational models of creativity that lack the ability to recognise the novelty of their works. Often computational models rely on generative mechanisms to produce 'interesting novelty' without any means of checking that this is the case other than referring to a human supervisor. Models of scientific and mathematical discovery include heuristics to guide their search processes by determining the interestingness of concepts but these have been shown to be inadequate in systems such as Lenat's AM which still required user interaction to produce creative works (Colton et al., 2000a). An objective of this work is to develop a general-purpose heuristic to guide the exploration of conceptual spaces in search of novelty.

A second motivation for this research comes from studies of creative designing showing the importance of reflecting upon work during the design process. Studies of designers at work have emphasised the interactive nature of designing. Schön calls this style of working 'reflection-in-action' and suggests that the processes involved are critical in many types of problem-solving (Schön, 1983). Models of designing based on Schön's studies place great emphasis on a designer's attendance to the emergence of unexpected consequences of design actions (Schön and Wiggins, 1992; Suwa et al., 1999). To model reflection-in-action design agents must be able to recognise unexpected consequences of their actions. An initial model of reflective sketching is given at the beginning of Chapter 5.

A final motivation comes from the view that creativity is a social-cultural construct, i.e. an honorific label assigned by peers and historians. In this view, creativity cannot be modelled as a closed system within a single agent: instead creativity must be modelled in the context of a society. Csikszentmihalyi[1] (1988; 1999) has been a vocal critic of computational models of creative thinking for not taking into account the effect that society has on the creative agent. The work presented towards the end of this thesis is an initial attempt to integrate some of Csikszentmihalyi's observations into an abstract computational framework for

[1] Pronounced "chicks-sent-me-high".

studying creativity. It is facilitated by the autonomy of curious design agents that permits the construction of artificial societies in which to situate creative activity.

1.2 AIMS AND OBJECTIVES

The aim of this research is to develop an understanding of the role that curiosity plays in creative thinking and creative design. To achieve this aim the following objectives were set:

1) To identify a computationally applicable notion of interestingness that captures what is meant when something is said to be novel or surprising.
2) To develop computational processes that can model interest and boredom based on this notion of interestingness.
3) To develop a model of curiosity that uses interest and boredom and that can be used to guide the actions of a design agent.
4) To develop a computational architecture for developing curious design agents that incorporates the processes of curiosity.
5) To investigate the behaviour of curious design agents in different situations for a number of design domains.

1.3 OVERVIEW OF THESIS

1.3.1 Background

Chapter 2 briefly reviews previous theoretical and computational models of creativity. The majority of Chapter 2 is spent examining the theoretical aspects of Berlyne's work on the perception of novelty and its effects on the behaviour of organisms. Different forms of novelty are described together with the related concepts of surprise, incongruity, and uncertainty and their relationship to expectations, conflict, and the perception of complexity. Berlyne's model of arousal and its relationship to judgements of aesthetics is also covered, importantly Berlyne's theory predicts that the most interesting novelty will be found in artefacts that are similar-yet-different to more familiar works. Berlyne's theory of curiosity is presented to differentiate the types of behaviours that can arise from the perception of novelty. Finally, some recent work developing computational models of curiosity is presented. Computational models have been developed to investigate the benefits of incorporating curiosity in autonomous learning systems, in particular, software agents and mobile robots.

1.3.2 A Computational Framework for Curious Design Agents

The first half of Chapter 3 describes a computational framework for developing curious design agents. The chapter begins with a functional description of a general-purpose agent framework. The addition of a curiosity module is shown to be a relatively small task requiring the modification of few existing processes. The second half of Chapter 3 describes the development of multi-agent simulations involving curious design agents and how this can be used to model Csikszentmihalyi systems view of creativity thanks to the insights of Liu (2000).

1.3.3 Implementing Curiosity

Chapter 4 provides some implementation details of the common components found in the curious design agents described later. In particular, the implementation and behaviour of the neural networks used to implement long-term memory are examined to give some indication of how they might affect curious agent behaviour. The essential process of novelty detection is described and two implementations are presented.

1.3.4 Applications 1: Curious Design Methods

The first chapter of applications, Chapter 5, demonstrates curious design agents in visual design domains using three different design methods: direct manipulation, parametric configuration and design tool-use. Each application is described in a separate section that begins with a brief account of the motivation to develop an agent for the chosen domain; continues with domain-specific implementation details; some experimental work to examine curious behaviour; and finishes with an application specific discussion of the results and potential directions for future work. The collection of agents described in this chapter shows that curiosity is a general-purpose search heuristic that can be applied to agents that design using different levels of abstraction in the design process.

1.3.5 Applications 2: Designing for Other Agents

Chapter 6 presents two more applications of curious design agents that examine different aspects of curiosity in design. The first application uses a simulation of crowd behaviour to examine the design behaviour of a curious agent in a non-visual domain. It also demonstrates the ability of a curious design agent to use the onset of boredom as a trigger to switch between problem-solving and problem-finding. The second application builds on the work done to develop a tool-using design agent in the previous chapter to develop a multi-agent simulation of a creative system, called The Digital Clockwork Muse, using the artificial creativity framework described in Chapter 3. Experiments with the simulation show the emergence of social definitions of whom and what are creative and the development of 'creative cliques' consisting of agents with similar notions of what is creative.

1.3.6 Discussion and Conclusion

The final two chapters discuss the possibilities for developing future curious design agents: additional functions for curious design agents are discussed; the possible application of curious design agents in CAD systems is examined; and the exciting promise of artificial creativity systems to provide insights into the social nature of creative designing is elaborated with some possible directions for future artificial creativity simulations.

Chapter 2

Background

2.1 WHAT IS CREATIVITY?

Creativity is the ability to produce work that is novel and appropriate.

There have been many attempts to be more specific than to define creativity as the ability to produce work that is novel and appropriate, Taylor (1988) gives some 50 definitions, but this simple statement appears to be the only definition upon which there is a consensus among the research community (Boden, 1990; Partridge and Rowe, 1994; Rosenman and Gero, 1993; Sternberg, 1988; 1999).

The purpose of studying creativity is often to determine what processes are involved in being creative or finding out what is meant when something is described as being novel (original, unexpected, surprising) and appropriate (useful, valuable, aesthetic, adapted).

2.1.1 Approaches to Studying Creativity

Many different approaches have been taken in studying creativity (Sternberg, 1988). Theoretical models have been proposed to provide more detailed accounts of the processes involved in creativity, e.g. Dewey (1910), Poincaré (1913), Wallas (1926), Guilford (1967), Koestler (1964), Hofstadter (1979); de Bono (1986), Martindale et al. (1988), Boden (1990), Finke et al. (1992), and Dacey and Lennon (1998).

Empirical studies of problem-solving and other creative tasks have been conducted to determine the characteristic traits of creative people and creative processes, e.g. Guildford (1967), Simonton (1997), Schön (1983), Sternberg (1988), Gardner (1993), and Csikszentmihalyi (1996a; 1996b). Amabile has conducted extensive studies of motivation in creative individuals and has found that intrinsic motivations play an essential role (1983; 1985; Collins and Amabile, 1999). Martindale's studies of the development of creative styles places the desire to find novelty in the unusual position of being one of the few constant motivations in creative work (Martindale, 1990).

Studies of preference judgements for creative artefacts have been provided insights into the nature of creative works, e.g. Berlyne (1971), Humphrey (1973), Whitfield and Wiltshire (1982), Gaver and Mandler (1987), Martindale et al. (1988), Martindale (1990). Many of these studies show that novelty plays a crucial role in the preference judgements of individuals, and that these preference judgements change with exposure to examples of a style.

2.1.2 Personal and Social Views of Creativity

Approaches to studying creativity can be divided into two broad categories. Firstly, there are those that emphasise personal judgements of creativity and study creative thinking and creative personalities. Secondly, there are those approaches that recognise that creativity goes beyond the individual and that society, as the audience of the creative work, plays an important role in defining what is creative.

The first approach to studying creativity has resulted in models of creative thinking (e.g. Wallas, 1926; Newell et al., 1962; Koestler, 1964; Martindale et al., 1988; Boden, 1990; Finke et al., 1992; Dacey and Lennon, 1998). Most computational models of creativity are based, either directly or indirectly, on these process models of creative thinking (e.g. Langley et al., 1987; Hofstadter et al., 1995a; Partridge and Rowe, 1994).

Proponents of the second approach contend that creativity cannot occur in a vacuum and must be studied in the context of the socio-cultural environment of the creator (Csikszentmihalyi, 1988; 1999). This definition has been popular in fields that consider the creativity of multiple individuals over extended periods of time, for example, in history, sociology and anthropology (Martindale, 1990).

2.1.2.1 Unified Models of Creativity

Some researchers have attempted to combine these two views of creativity into unified theoretical frameworks but the resulting frameworks often maintain the distinction between personal and socio-cultural notions of creativity, as with Gardner’s small-c and big-c creativity (Gardner, 1993) and Boden’s P-creativity and H-creativity (Boden, 1990).

Boden (1990) classifies creativity as either *historical creativity* (H-creativity) or *psychological creativity* (P-creativity). P-creative ideas are novel with respect to an individual’s experiences; H-creative ideas are novel with respect to the whole of human history. H-creativity presupposes P-creativity, if an individual has an idea that is historically novel then it must also be novel to that individual as well as to others. Gero has extended Boden’s classification to include *situated creativity* (S-creativity). S-creative ideas are novel with respect to the situation of an individual emphasising the important role that context plays in shaping the creative process (Suwa et al. 1999).

2.1.2.2 A Systems View of Creativity

Csikszentmihalyi (1988) presented a different approach to studying creativity by considering the interactions between individuals, society and culture. When Csikszentmihalyi developed his systems view of creativity, he turned his attention away from the question "What is creativity?" and focussed upon the issues surrounding the question "Where is creativity?" Importantly, Csikszentmihalyi questioned the mentalistic assumption that creative processes are only to be found in the mind of the creative individual. Instead he proposed that processes essential to creativity, whether personal or socio-culturally defined, are to be found in the interactions between individuals and the society that they are situated within.

The systems view of creativity was developed by Csikszentmihalyi as a model of the dynamic behaviour of creative systems that include interactions between the major components of a creative society (Csikszentmihalyi, 1988). Csikszentmihalyi identified three important components of a creative system; firstly there is the *individual*, secondly there is a cultural, or symbolic, component called the *domain*, and thirdly there is a social, or interactive, component called the *field*. A map of the systems view of creativity is presented in Figure 2.1.

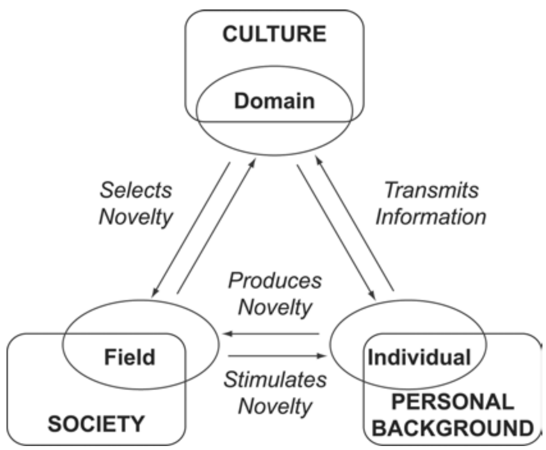


Figure 2.1: Csikszentmihalyi's systems view of creativity (after Csikszentmihalyi, 1999).

An individual's role in the systems view is to bring about some transformation of the knowledge held in the domain. The field is a set of social institutions that selects from the variations produced by individuals those that are worth preserving. The domain is a repository of knowledge held by the culture that preserves ideas or forms selected by the field.

In a typical cycle, an individual takes some information provided by the culture and transforms it, if the transformation is deemed valuable by society, it will be included in the domain of knowledge held by the culture, thus providing a new starting point for the next cycle of transformation and evaluation. In Csikszentmihalyi's view, creativity is not to be found in any one of these elements, but in the interactions between them.

2.2 COMPUTATIONAL MODELS OF CREATIVITY

Developing computational models of creativity is a relatively recent approach that is closely associated with artificial intelligence (AI) and cognitive science (Langley et al., 1987; Partridge and Rowe, 1994; Boden, 1999). Developing computational models of creativity can potentially do three things (Elton, 1995):

1) Produce computational systems that produce novel and appropriate works, e.g. scientific theories, musical compositions, architectural designs etc.
2) Contribute to the cognitive sciences as it seeks to understand the mechanisms involved in human creative thinking, e.g. analogy-making, emergence etc.
3) Provide abstract models of creativity that are not tied to a specific domain or process for the study of creativity in its most general sense.

In accordance with the definition of creativity given earlier, computational models of creativity are often assessed against the first of these goals: the ability to produce novel and appropriate works. In some cases the performance of computational systems has been measured against works that have been considered creative in the past (Langley et al., 1987), in other cases the creativity has been assessed against the state-of-the-art in a domain (Lenat, 1983).

2.2.1 Computational Approaches

This section presents some successful computational systems to illustrate some different computational approaches to modelling creativity.

2.2.1.1 Grammar-Based Systems

Rule-based systems have been a popular way to encode knowledge in both AI and design computing. A carefully constructed set of rules constitutes a grammar that can guarantee the production of appropriate works within a style. For example, take the shape grammar for generating Frank Lloyd Wright's Prairie Houses (Koning and Eizenberg, 1981). The Prairie House grammar was constructed by careful study of Frank Lloyd Wright's designs and extracting the complex relationships between design elements as rules. The grammar could then be used to generate new houses in the same style as Lloyd Wright. The results of such a system are often excellent examples of the style, but they are formulaic; they lack the originality that a creative talent like Lloyd Wright would introduce. There is nothing in the system of rules that can motivate the shift from Prairie Houses to Waterfall to a Guggenheim Museum.

Schnier and Gero have produced evolutionary design systems (see Section 2.2.1.3) that can learn the style of Lloyd Wright's buildings by identifying the 'building blocks' that define the style, thus avoiding the need to hand craft a production grammar (Schnier and Gero, 1996; Schnier, 1999). They have used these systems to generate highly innovative designs, as they did when they combining learned patterns for Frank Lloyd Wright window designs with those learned for Mondrian paintings – producing original Flondrians (Schnier, 1999). But as before, the motivation for this undoubtedly creative combination came from the developers and not from any intrinsic motivation of the system to explore new possibilities.

2.2.1.2 Discovery Systems

Some of the earliest computational models of creativity were developed as models of scientific and mathematical discovery. The history of discovery systems stretches back to the beginnings of AI with Newell and Simon's pioneering work developing the *Logic Theorist* (Crevier, 1993). The 1960's saw the development of the first expert system for scientific hypothesis formation: DENDRAL (Lindsay et al., 1993) and in the 1980's a milestone was reached with the BACON family of scientific discovery programs (Langley et al., 1987).

Simon claimed that BACON family of programs should be considered creative because they could replicate the solutions to some of the most creative problems in science, but Csikszentmihalyi has argued that these programs are more like forgers and that as such they cannot be considered creative (Csikszentmihalyi, 1988). Csikszentmihalyi's argument emphasises the need for a creative system to produce novel works and not just reproduce previously novel works because in many of the cases of scientific discovery, finding the right problem, or asking the right questions, is more important than finding solutions (Einstein and Infeld, 1938). Csikszentmihalyi argues that because the discovery systems were developed with the intention of solving a specific problem and given all of the necessary knowledge to do so, they cannot claim the creativity of the original discoveries.

The beginning of the 80's also saw Lenat develop the Automated Mathematician (AM). Lenat (1976) claimed that AM was creative because it developed several new and interesting concepts, however, the creativity of AM is also problematic (Lenat and Brown, 1984; Ritchie and Hanna, 1984; Rowe and Partridge, 1993; Boden, 1999). Despite having many heuristics dedicated to determining the interestingness of concepts, AM would produce many concepts that human mathematicians would consider uninteresting and it required user intervention to identify the most promising directions for exploration, as Lenat himself noted "the very best examples of AM in action were brought to full fruition only by a human developer" (Davis and Lenat, 1982).

Lenat attributed the limitations of AM to the fixed nature of its heuristics (Lenat and Brown, 1984) and his later work developing EURISKO tried to address this by using meta-heuristics to generate new heuristics as needed. EURISKO had some considerable success and produced several innovations including one that was awarded a U.S. patent (Boden, 1999). Despite its successes, EURISKO has not been widely used, largely because it relied upon many domain-specific heuristics that required considerable effort to produce.

Work has continued to develop scientific and mathematical discovery systems, and new approaches, like Inductive Logic Programming (ILP), have seen real success when applied to narrowly defined domains (Colton and Steel, 1999). One of the most exciting developments has been that of closed-loop discovery systems, as originally demonstrated by Hayes-Roth (1983). In closed-loop systems, a computer is provided with the means to plan and perform experiments to test hypotheses without human intervention (e.g. Bryant et al., 1999). These systems show one way forward for developing autonomous models of discovery, but one that has as much to do with solving domain-specific problems as studying the nature of creativity.

2.2.1.3 Generate-and-Test Systems

One of the most common approaches to modelling creativity is to use one generative function to produce a number of solutions and then select the best from that set using an evaluation function: implementing a generate-and-test cycle similar to the incubation and illumination processes of Wallas' theoretical model (1926). Typically, computational models of creativity that use a generate-and-test cycle break the task of producing potentially creative products into two: (1) generate novel products, and (2) test products for appropriateness.

The generate-and-test approach is typified by use of evolutionary systems in art and design. Evolutionary algorithms have been applied with great success to a wide range of creative design problems (Bentley, 1999d). Evolutionary design systems use a set of genetic operators to generate genetic representations that are expressed into designs and evaluated for their usefulness using a fitness function. Evolutionary systems have been used in many design applications including the design of electronics (Koza et al., 1999; Thompson, 2000), mechanical systems (Eby et al., 1999), buildings (Rosenman, 1996; Bentley, 1999c), furniture (Bentley, 1999c), artworks (Sims, 1991; Todd and Latham, 1992; Witbrock and Reilly, 1999), and even artificial lifeforms (Lipson and Pollack, 2000). The success of evolutionary design systems has resulted in some researchers speculating that they model creativity (e.g. Goldberg, 1999), although most commentators are justly cautious and do not make such claims without some reservations (e.g. Bentley, 2000).

Connectionist systems implementing a generate-and-test model of creativity have also had considerable success (Thaler, 1996). Boden (1990) suggests that connectionist systems have several qualities, such as robustness in the face of noise, that make them a good choice for developing computational models of creativity. Other commentators appear to agree that neural networks offer a level of flexibility that is beneficial (e.g. Clark, 1994).

2.2.2 Criticisms of Computational Models

Despite the success of the above systems, several researchers have argued that these systems do not model creativity because they cannot successfully recognise the creativity of their own work (Csikszentmihalyi, 1988; Boden, 1990; Elton, 1995). In Elton's view, meaningful evaluation is all-important in the assignment of creativity, he states that: "Generation, however masterful, without evaluation just does not count as creativity." Elton's solution to the problem is to develop computational systems that have enough cultural knowledge that they can evaluate the novelty of their own work to determine whether it counts as being creative (i.e. in the sense of Boden's P-creative).

2.3 NOVELTY

To develop design agents that can evaluate the creativity of their works we need to develop the means to evaluate the novelty of those works. The subjective evaluation of novelty is quite different from that of usefulness: a creative product is likely to remain useful for some time but it loses its novelty as soon as it is experienced. This makes the application of fixed heuristics to determine novelty inappropriate, as was demonstrated by the continued reliance of AM and EURISKO on human assistance.

Berlyne conducted extensive research into the effects of perceiving novelty on the behaviour of humans and animals (Berlyne, 1960) and their roles in the judgement of aesthetics (Berlyne, 1971).

This section presents a very brief introduction to the work of Berlyne and other experimenters interested in studying novelty and related concepts of interestingness, and curiosity. Berlyne's theories of novelty are presented first before examining some previous attempts to model curiosity in computational systems. Berlyne identified several dimensions for differentiating types of novelty that capture on the comparative, temporal, and epistemic properties of novel stimuli.

2.3.1.1 Dimensions of Novelty

One of the most important distinctions between different types of novelty is between novelty that is due to atypical stimuli and novelty due to a stimuli being uncommon:

- *Atypical stimuli* are unlike previous experiences. The novelty of an atypical stimulus lies in the differences between it and the nearest matching previous experiences, or the improbability of its appearance given previous experiences of similar stimuli.
- *Uncommon stimuli* are familiar from previous experiences but are rarely experienced or have not been experienced for some time. The novelty of an uncommon stimulus comes from the improbability of the experience.

Another difference between forms of novelty is the timeframe over which the novelty is detected. Berlyne (1960) distinguishes between immediate, short-term and long-term novelty:

- *Immediate novelty* is the novelty of an experience at an instance. The immediate novelty of a stimulus, such as a visual pattern, is based on the elements present in the sensory field.
- *Short-Term novelty* is the novelty of an experience relative to recent experiences. The short-term novelty of a sequence of stimuli, such as a melody, is based on the contents of short-term memory.
- *Long-Term novelty* is the novelty of an experience relative to experiences that may be hours, days or years old but that have left a trace in long-term memory.

In addition Berlyne (1960) differentiates between perceptual and epistemic forms of novelty:

- *Perceptual novelty* is defined in relation to perceptions. The perceptual novelty of a stimulus is based on the comparison of non-symbolic properties of the stimulus.
- *Epistemic novelty* is defined in relation to knowledge. The epistemic novelty of a stimulus is based on a comparison of the meaningful associations that it connects with in the mind of the observer.

To clarify, perceptual novelty is the sort of novelty that draws one's attention to a stimulus without requiring the stimulus to be identified as something of a particular type. Epistemic novelty requires that the stimulus be recognised as being of a certain type before the novelty can be appreciated.

2.3.2 Degree of Novelty

Intuitively, the degree to which a stimulus pattern is novel will be inversely proportional to:

1) How often similar patterns have been experienced.
2) How similar these patterns have been.
3) How recently these patterns have been experienced.

Computationally, novelty is detected using processes that estimate one or more of these properties for a given stimulus pattern and a representation of previous stimuli.

2.3.3 Novelty Related Concepts

Some concepts are naturally related to novelty, in some cases so much so that they are commonly considered synonymous. These related concepts are worth considering separately as they provide different opportunities for generating interesting novelty in creative works and require different mechanisms to recognise them.

2.3.3.1 Surprise

A surprising stimulus is not just atypical or uncommon; it is a stimulus that disagrees with one or more expectation (Berlyne, 1960). Surprise involves anticipation of an experience that is not fulfilled by the actual experience that follows. The degree of surprise depends upon the confidence put in the expectation and the degree to which the expectation is confounded. As Berlyne (1971) notes, it is sometimes hard to convince people that surprisingness is not the same thing as novelty, but they are distinct and something can be surprising without being novel and vice versa. The difference between novelty and surprise is nicely illustrated by the following extract written over 200 years ago:

> … an elephant in India will not surprise a traveller who goes to see one; and yet its novelty will raise his wonder: an Indian in Britain could be much surprised to stumble upon an elephant feeding at large in the open field; but the creature itself, to which he was accustomed, would not raise his wonder.
>
> (Home, 1795; quoted in Berlyne, 1971 pp. 146)

2.3.3.2 Incongruity

When a stimulus sets up an expectation that is not satisfied within the same experience, it is said to be incongruous rather than surprising. The difference between surprising and incongruous experiences is blurred by the sequential nature of many sensory experiences, such as the scanning of visual patterns. An incongruous stimulus can be constructed by substituting an element of a familiar stimulus with one that is unexpected or unfamiliar.

2.3.3.3 Uncertainty

Uncertainty arises when there is no clear response to a stimulus. Stimuli that cause uncertainty may either be too distant from familiar experiences to be classified with confidence or may evoke multiple responses equally by falling somewhere between them. The concepts involved in uncertainty are likely to be similar. For example, consider trying to identify a shade of colour that is a mixture of red and pink, the result is likely to be uncertain, resulting in descriptions such as "reddish-pink" or "pinkish-red", neither of which give a clear indication of the actual colour.

2.3.4 Conflict

Conflict is a super-ordinate concept related to surprise, incongruity and uncertainty, capturing the common pattern of response found in all three (Berlyne, 1971, pp. 150). Conflict is caused by a stimulus simultaneously evoking multiple responses that compete for dominance. The degree of conflict will depend upon the confidence of the competing responses.

2.3.5 Expectations

Expectations obviously play an important role in the perception of surprising and incongruous stimuli. Expectations can be formed in three main ways. The most common way is through the repeated experience of combinations of stimuli or sequences of events. Classical conditioning then leads the perception of a stimulus, X, to evoke an expectation of a response that usually accompanies or follows it, Y. The strength of the expectation will reflect the reliability with which Y can be predicted given X, i.e. $p(Y|X)$. Secondly, expectations of something can be formed on the advice of a reliable information source. Finally, expectations can be formed through a reasoning process.

2.3.6 Complexity

In general, the more novelty a stimulus presents the more complex it will appear to be, however, complexity does not imply novelty: a highly complex stimulus, such as a natural scene or an intricate design, is not necessarily novel. Martindale (1990) proposed that the continuous search for novelty in art and design styles would naturally lead to an increase in the complexity of works within the style over time. According to Martindale, stylistic changes happen when the complexity of works within a style become too high to be intelligible at which point a new style has the chance to become popular as a simpler, yet novel, alternative.

2.4 INTERESTINGNESS

What's interesting? Schmidhuber (1997) answers this question by writing "Interestingness depends on the observer's current knowledge and computational abilities. Things are boring if either too much or too little is known about them – if they appear either trivial or random." Interestingness is obviously closely tied with learning and this can explain why truly creative works often require great efforts on the part of the creator to educate the potential audience in order for them to appreciate the creativity of the work (Csikszentmihalyi, 1988).

2.4.1 Objective and Subjective Notions of Interestingness

Silberschatz and Tuzhilin (1996) suggest that interestingness can be either objective or subjective: objective interestingness uses relationships found entirely within the object considered interesting, while subjective interestingness compares properties of the object with beliefs of a user to determine interest. Two aspects that make something subjectively interesting are that it is unexpected and/or actionable. Unexpectedness depends upon an agent's ability to predict an as-yet-unseen event. Actionability depends upon whether an agent can take action as a consequence of a discovery. The two concepts are conceptually independent although Silberschatz and Tuzhilin note that unexpected discoveries are often actionable. Silberschatz and Tuzhilin's definitions of interestingness are compatible with Berlyne's definitions of novelty and surprise. This work is concerned with subjective notions of interestingness in creative design: an interest in the unexpectedness of novel artefacts and the curious design actions that result.

2.4.2 Hedonic Value

Interest can be considered a special case of "hedonic value" associated with the pleasure associated with heightened states of learning. To describe the response to arousing stimuli, Berlyne (1971) coined the term "hedonic value", in reference to the pleasure/pain response that is often associated with arousal. Berlyne's model of the relationship between arousal and hedonic value uses a non-linear function that resembles an inverted U-shape called the Wundt curve (Berlyne, 1971). The Wundt curve is name after the pioneering experimental psychologist Wilhelm Wundt; it is sketched in Figure 2.2.

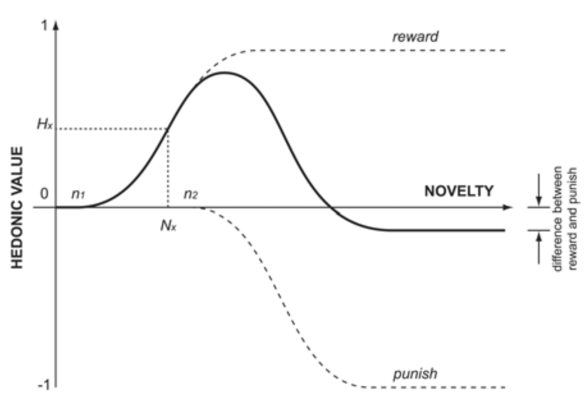


Figure 2.2: The Wundt Curve: a hedonic function used to calculate interest. The hedonic function is shown as a solid line, the reward and punishment sigmoidal curves summed to form the hedonic function are shown dashed.

Importantly, the maximum value of the Wundt curve is often located in a narrow region close to zero. In the case of interestingness, the shape of the Wundt curve means that the most interesting experiences are those that are *similar-yet-different* to those that have been experienced previously. Berlyne's model is supported by empirical evidence gathered from studies of aesthetic preference and creative thinking. Berlyne proposed that his model of arousal is also the basis of behaviour commonly referred to as 'curiosity'.

2.5 CURIOSITY

Berlyne (1971) defines curiosity as a form of motivation that promotes exploratory behaviour to learn more about a source of uncertainty, such as a novel stimulus, with the goal of acquiring sufficient knowledge to reduce the uncertainty.

2.5.1 Exploratory Behaviour

Berlyne (1971) presented two types of motivations for exploration, diversive and specific, in keeping with his model of hedonic reward. In diversive exploration, an organism is under-stimulated and seeks arousal from the environment. In specific exploration, an organism is over-stimulated and seeks to reduce its arousal by reducing the novelty of the situation and its associated collative variables, in particular, uncertainty.

Berlyne (1960) proposed three mechanisms of exploratory behaviour observed in higher mammals: orientation, locomotion and investigation.

- *Orientation*: An animal will orient itself towards an object of interest in order to gain more information about it.
- *Locomotion*: An animal will move towards an object of interest in order to gain more information.
- *Investigation*: An animal will affect changes in an object of interest to gain information.

In design, we are mostly interested with the third type of exploratory behaviour, where new designs are developed to investigate imagined possibilities motivated by curiosity in an existing design, however, the first two types of exploratory behaviour may also be important, especially in the early stages of design when little is known about a design problem and useful information can be gained more readily from inspecting existing designs than from generating new ones.

In any case, whether motivated by diversive or specific needs, and implemented through orientation, locomotion or investigation, the goal of such exploratory behaviour is to gain knowledge and this typifies curious behaviour. As Schmidhuber (1991b) comments: "One gets curious as soon as one *believes that there is something that one does not know*." As this statement can include both diversive and specific motivations, if one makes the reasonable assumption that there is always something more to learn.

2.5.2 Agents

Agents have become a popular vehicle for artificial intelligence (Franklin, 1997a; 1997b). Although a precise definition of what constitutes an agent remains elusive

(Wooldridge and Jennings, 1995; Franklin and Graesser, 1996; Nwana, 1996; Wooldridge, 1999), one aspect of agency that is generally agreed upon is that agents are autonomous, i.e. that they can operate without assistance in complex environments.

The level of autonomy is an important topic in agent research. In the last decade agent research has shown that representations grounded in the experience of an agent are more useful for dealing with complex environments. Agents that develop their own representations of the world in terms of their interactive experiences are called situated, they go beyond the operational autonomy found in other agents and gain a level of epistemic autonomy (Ziemke, 2000). Curious agents need to situate their experiences of design domains so that they can determine the novelty of new design products.

2.5.2.1 Curious Agents

Schmidhuber (1991b) discussed the possibilities for implementing curiosity and boredom in model-building neural controllers for autonomous agents. Schmidhuber has demonstrated with a number of autonomous agents that engage in self-directed learning in complex environments that curiosity can be very effective in guiding the exploration of dynamic environments (Schmidhuber, 1991a; 1991b; 1991c; 1997). Schmidhuber implemented curious agents using neural controllers and reinforcement learning with intrinsic rewards generated in response to an agent improving its model of the world (Schmidhuber, 1991c). Curious agents are rewarded for learning to predict aspects of the world with greater accuracy so that it can better predict the consequences of its actions.

One of the most interesting curious agents developed by Schmidhuber used two "brains" in competition to determine the most interesting course of action at an instant (Schmidhuber, 1997). Each brain conducts independent experiments to explore a space of possibilities. When a brain discovers a surprising result, it can challenge the other brain to predict the result in an attempt to surprise it. The other brain has the ability to veto the challenge if it is not confident that it can predict the result. If a brain accepts a challenge and loses, it is punished and the challenger rewarded. If the challenged brain wins it is rewarded and the challenger punished. The result of using a two-brain approach is robust agent behaviour that balances exploration versus exploitation and was shown to outperform standard reinforcement learning techniques set the task of exploring a simulated environment.

Marsland et al. have developed robots that display orienting and locomotive exploratory behaviour motivated by curiosity that they call *neotaxis* (Marsland et al., 2000a; 2000b; 2000c; 2001). The habituated mechanisms used by the robots to detect novelty respond to how recently an input was last experienced. The robots have been shown to detect novel features of an environment that aid the efficient exploration of complex environments that are initially unknown.

Macedo and Cardosa (2001a, 2001b) presented a model of surprise and curiosity in a design system using Case-Based Reasoning. They have presented their model as general-purpose design search heuristic in much the same way that the one described in the following chapter is presented. Although they describe their model as one of surprise, it is closer to Berlyne's definition of novelty because the expectations used are implicit to the learning process, rather than explicitly generated in preparation for an anticipated experience. A more detailed discussion of the work of Macedo and Cardosa is presented in Chapter 7.

2.5.2.2 Attentive Agents

The development of active vision systems has become an important topic in robotics and other applications that must handle visual data (Aloimonos et al., 1988). Some researchers have approached the problem of determining what to pay attention to by developing models of selective attention based on expectations, novelty and surprise; implementing forms of perceptual curiosity (Baluja, 1996; Peters, 2000).

2.6 CONCLUSION

The background research presented in this chapter has provided three important insights that have shaped the work presented in the remainder of this thesis:

1) Relatively little work has been done to develop processes for recognising novelty in computational models of curiosity.
2) A body of work exists in psychobiology and related fields that provides theories and models of the novelty detection, the determination of interestingness, and curiosity in organisms.
3) Computational models of novelty detection, and curiosity exist and have been implemented in agents applied to domains other than design.

The evaluation of novelty plays a crucial role in determining the creativity of products and of the processes or people that produce them. It is therefore somewhat surprising that computational models of creativity have generally relied upon generative processes to produce interesting novelty without an explicit test. This is especially true given the number of different ways that novelty can be experienced, e.g. surprise, incongruity etc., and the non-linear relationship between novelty and interestingness. Existing models of curiosity suggest that curious design agents can be developed that can autonomously explore a design space and situate their design actions in their history of experiences.

The importance of Berlyne's work cannot be overstated: Berlyne's model of hedonic reward for arousal-stimulating devices, like novelty, has been a cornerstone of modern research in aesthetics and fundamental in the development of the model of curiosity presented here. The explicit application of Berlyne's model of hedonic reward to the output of novelty detecting mechanisms represents one of the main differences between the model of curiosity developed in the following chapters and those developed by others.

Chapter 3

A Computational Framework for Curious Design Agents

The previous chapter has provided the background needed to identify the necessary components of a curious design agent. Curious design agents need to interact with an environment, learn from those interactions, determine their interest in new experiences relative to a history of previous experiences and take actions to promote further interesting experiences.

The aim of this chapter is to provide a framework for describing curious design agents. The objectives are (1) to describe the functional components that make up a curious design agent, and (2) to present an approach to developing multi-agent simulations using curious design agents.

This chapter begins with a description of the components of an agent as an abstract architecture that delineates the functions of sensing, perceiving, conceiving, acting, effecting, and remembering. A framework for curious design agents is presented, adding the components necessary to support curiosity. The model of curiosity presented here requires functional units to be added for novelty detection, interestingness evaluation, and curious concept formation. The second half of this chapter develops a framework for developing multi-agent simulations of creative societies based on Liu's dual generate and test model of social creativity (Liu, 2000).

3.1 A SIMPLE AGENT

The abstract architecture for a simple agent presented in this section is based on the framework presented by Wooldridge (1999). The abilities of curious design agents and their components are given as functions that map between states of different types. Functions maintain no state information across time and so the requirements for short-term and long-term memory for an agent are described separately, providing a clean separation of function, state and memory. The description of a simple agent begins with the environment with which the agent interacts. The internal components of a simple agent consist of processes for sensing, perceiving, conceiving, action, effecting as well as short-term and long-term memory.

3.1.1 Environments

An agent's environment can be characterised as a set, W, of all possible environment, or world, states:

$$W = \{w_0, w_1, \ldots\} \tag{3.1}$$

3.1.2 Agents

An agent's ability to effect change in the world is determined by the range of its effectors that is assumed to be characterised by a set, E, of all possible states that an agent's effectors can be configured to take:

$$E = \{e_0, e_1, \ldots\} \tag{3.2}$$

Consequently, an agent can be viewed in its most abstract sense as a function, $\boldsymbol{\alpha}$, that maps sequences of environment states to new environment states:

$$\boldsymbol{\alpha} : W^* \to E \tag{3.3}$$

At a given time an agent decides on an action to effect, $e \in E$, based on its history of experiences of its environment to date, W^*. The processes involved in an agent deciding to take an action are described below.

The behaviour of the environment with respect to an agent's actions can be represented as a function, $\boldsymbol{\varepsilon}$, that maps the current environment state, $w \in W$, to the subset of possible environment states, $\boldsymbol{\mathcal{E}} \subseteq W$, after the agent $\boldsymbol{\alpha} : W^* \to E$ has effected it chosen action, $e \in E$:

$$\begin{gathered} \boldsymbol{\varepsilon} : W \times E \to \boldsymbol{\mathcal{E}} \\ \text{where} \\ \forall i, i > 0, w_i \in \boldsymbol{\varepsilon}(w_{i-1}, e_{i-1}) \\ \boldsymbol{\mathcal{E}} \subseteq W \end{gathered} \tag{3.4}$$

If the range of $\boldsymbol{\varepsilon}$ contains only singletons, i.e. sets containing only one member, then the environment is deterministic and the behaviour of the environment can be accurately predicted. Otherwise the environment is non-deterministic and the results of taking actions cannot be guaranteed, only estimated with some confidence.

3.1.2.1 Sensing

At the interface between an agent and its environment are its sensors and effectors. Sensors transform aspects of the external state of the world into a set of internal variables called *sense data*. The range of an agent's senses can be characterised as a set, S, of sensory states:

$$S = \{s_0, s_1, \ldots\} \tag{3.5}$$

An agent's sensory abilities are represented as a function, $\boldsymbol{S}$, that takes the current state of the environment, $w \in W$, and maps it to a sensory state, $s \in S$:

$$\boldsymbol{S} : W \to S \tag{3.6}$$

3.1.2.2 Perceiving

Perception is the process of extracting useful features, i.e. percepts, from the raw sense data. An agent's ability to perceive can be characterised by a set, P:

$$P = \{p_0, p_1, \ldots\} \tag{3.7}$$

The process of perception is represented as a function, $\boldsymbol{P}$, that takes the current sensory state of the agent, $s \in S$, and maps it to a perceptual state, $p \in P$:

$$\boldsymbol{P : } S \to P \tag{3.8}$$

3.1.2.3 Conceiving

The term conception is used here to encompass all of the high-level functions of an agent, including belief revision, goal setting and plan setting. An agent's ability to take appropriate actions is determined by its ability to conceive of its situation and can be characterised by a set, C, of conceptual states:

$$C = \{c_0, c_1, \ldots\} \tag{3.9}$$

An agent's decision-making processes are represented as a function, $\boldsymbol{C}$, that maps sequences of perceptual states to a conceptual state:

$$\boldsymbol{C : } P^* \to C \tag{3.10}$$

Note that the conceptual processes are the only ones that take a sequence as an input and the consideration of an agent's situation is localised in the highest-level of agent function.

3.1.2.4 Acting

Action is the process of translating high-level goals and plans lower-level commands that can be carried out by effectors. The actions that an agent takes are represented as a set, A:

$$A = \{a_0, a_1, \ldots\} \tag{3.11}$$

The process of determining which action to take is represented as a function, $\boldsymbol{A}$, that maps the current conceptual state, $c \in C$, to an action, $a \in A$:

$$\boldsymbol{A : } C \to A \tag{3.12}$$

3.1.2.5 Effecting

The effect of taking an action in an environment can be modelled as a function, $\boldsymbol{E}$, that takes an action, $a \in A$, and configures the agent's effectors it into new state, $e \in E$, to accomplish the chosen action:

$$\boldsymbol{E : } A \to E \tag{3.13}$$

3.1.2.6 An Abstract Agent

The above description represents all of the required functions for an agent. A description of an agent consisting of processes for sensing, perceiving, conceiving, acting and effecting can be written as a compound function:

$$\begin{gathered} \boldsymbol{\alpha :} W^* \xrightarrow{sense} S^* \xrightarrow{perceive} P^* \xrightarrow{conceive} C \xrightarrow{act} A \xrightarrow{effect} E \\ \boldsymbol{\alpha :} \boldsymbol{E}\left(\boldsymbol{A}\left(\boldsymbol{C}\left(\boldsymbol{P}\left(\boldsymbol{S}\left(W^*\right)\right)\right)\right)\right) \end{gathered} \tag{3.14}$$

The architecture for an abstract agent is illustrated in Figure 3.1. The functions described above as processes illustrated by circular nodes. There may be several processes contributing to a single function in an agent as in the case of the sensors and effectors. Solid arrows represent the flow of state variables between processes.

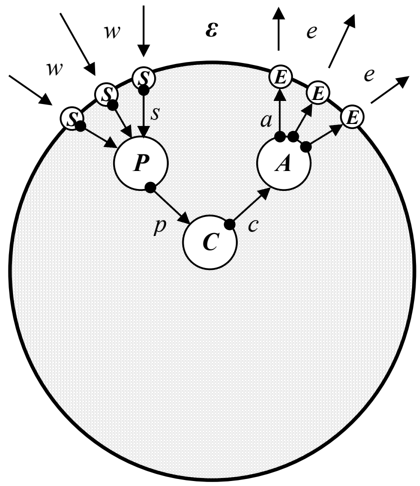


Figure 3.1: A simple agent architecture.

Simple agents of the type illustrated in Figure 3.1 are not sufficient for developing models of curiosity. Curiosity is all about trying to learn about the world and such simple agents do not provide the necessary functions to learn as they interact with the world.

So far the decision functions of agents have been described as a mapping from sequences of environment states to an effected action. For the standard agent this is the agent function, $\boldsymbol{\alpha}: W^* \to E$, taken as a whole. In the simple agent described above the mapping from a sequence of input states is limited to the conceptual function, $\mathbf{C}: P^* \to C$, that maps from sequences of percepts to a concept. The use of this scheme allows agents to be represented whose decision-making is influenced by history. This is important because curiosity is all about trying to learn about the world and agents must therefore provide the necessary functions to learn as they interact with the world. Unfortunately, this is a rather unintuitive representation; in the following sections it will be replaced by an equivalent and more natural representation by considering agents with memory.

In practical terms, the minimal agent architecture to support curiosity must include some form of memory and a way of adapting that memory to store new experiences. Unfortunately, agents cannot be equipped with infinite memories using unlimited storage capacity, consequently, some assumptions must be made about the importance of previous experiences to determine what should be remembered. Two reasonable assumptions are that (a) recent experiences are likely to be the most relevant at the current time, and (b) similar experiences from any time in the past are likely to be useful in determining what, or perhaps what not, to do. These assumptions suggest two different ways of reducing the amount of space required to store representations of previous experiences:

1) Store representations of recent experiences.

2) Store generalisations of similar experiences.

Short-term memory stores a limited set of accurate representations of recent experiences. Long-term memory stores generalisations of more experiences from longer ago but with an increased chance that they may be recalled inaccurately due to the process of generalisation.

The amount of short-term memory an agent has may vary — but the need for short-term to store accurate descriptions of recent events will always mean that short-term memory will extend backwards in time for a relatively short period of time compared to the average lifespan of an agent. In humans, short-term memory is often said to contain 7±2 "chunks", where a chunk is taken to be an easily remembered item, e.g. a single letter, a word or a common phrase. Similarly, a computational agent might be expected to accurately remember a small number of items, e.g. 10, in short-term memory.

3.1.2.7 Short-Term Memory

A short-term memory unit, $\boldsymbol{STM}$, stores a number of percepts, actions and concepts for a short period of time to support communication between processes, time-based processing of inputs, and decision-making. The content of $\boldsymbol{STM}$ is represented as a set of recent percepts, actions and active concepts.

$$\begin{aligned}&\boldsymbol{STM:} P\times C\times A\times STM \to STM\\&STM=\{P,C,A\}\end{aligned} \tag{3.15}$$

To use $\boldsymbol{STM}$ the perception, conception and action processes need to be updated to store and retrieve variables. The updated process of perception, $\boldsymbol{P}$, takes sensed data, $s\in S$, and the contents of short-term memory, STM, and produces a new perceptual state, $p\in P$. Conception takes the contents of short-term memory, $\boldsymbol{STM}$, and produces a new conceptual state, $c\in C$. Action takes the contents of short-term memory, $\boldsymbol{STM}$, produces a new action, $a\in A$.

$$\begin{aligned}&\boldsymbol{P:} S\times STM \to P\\&\boldsymbol{C:} STM \to C\\&\boldsymbol{A:} STM \to A\end{aligned} \tag{3.16}$$

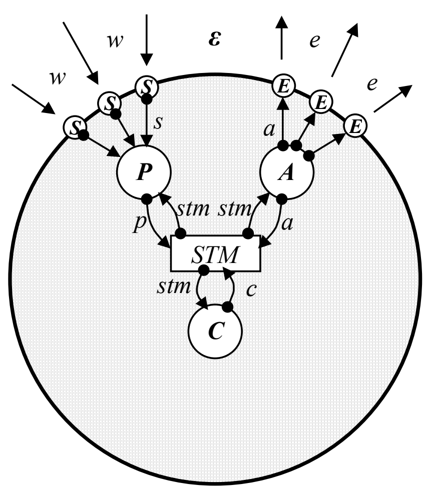


Figure 3.2: An agent with short-term memory.

An agent with short-term memory is illustrated in Figure 3.2. The figure shows that $\boldsymbol{STM}$ is located between the processes of perception, conception and action.

$\boldsymbol{STM}$ mediates the communication of variables between processes and allows a process to store its own state for future reference.

3.1.2.8 Long-Term Memory

Long-term memory stores representations of previous experiences. Long-term memory is represented as a function, $\boldsymbol{LTM}$, that takes the current conceptual state of the agent, $c \in C$, and maps it to a memory, $m \in M$, and updates its internal state as a side effect:

$$\boldsymbol{LTM} : C \to M \times LTM \tag{3.17}$$

Conceptual processes that make use of long-term memory map the remembered state, M, together with the content of short-term memory, to a new conceptual state, C.

$$\boldsymbol{C} : M \times STM \to C \tag{3.18}$$

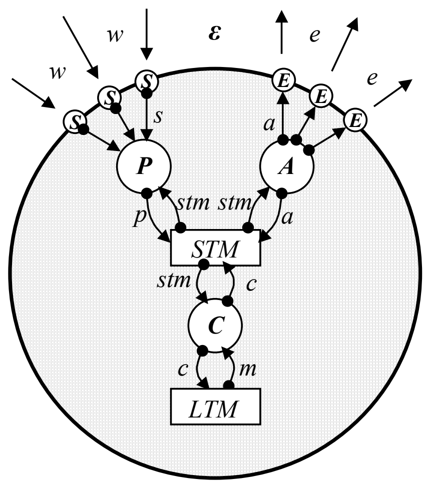


Figure 3.3: A learning agent with long-term memory.

3.2 A CURIOUS AGENT

The architecture for curious design agents adds meta-level conceptual functions to the architecture for agents already described. The meta-level functions determine the novelty of the current conceptual state of the agent, calculate the interestingness of the situation based on the novelty detected and set new goals for the agent based on the interestingness of past and present situations and the expected interestingness of future situations.

One of the goals of a model of curiosity is to determine aspects of the environment, as conceived by the agent, that are not modelled sufficiently well by $\boldsymbol{LTM}$. This is done by determining the novelty of a situation that indicates the atypicality of the current environmental state with respect to the agent's experiences. A second goal of curiosity is to identify unexpected consequences of actions. Again this involves identifying deficiencies in the model of the world held by $\boldsymbol{LTM}$, but in this case it is the dynamic nature of interaction that is under scrutiny. Curiosity in unexpected consequences is based on the detection of surprise that indicates that the predictions made by an agent about futures states of the environment have been proved incorrect.

3.2.1 Novelty and Surprise Detection

Novelty detection determines the novelty of a situation. Novelty detection is considered a meta-level conceptual process because the concept of novelty is based upon the concepts built by other conceptual processes categorising the situation. The novelty detection function, $\boldsymbol{N}$, takes the conceptual state of the agent, $c \in C$, and compares it with memories of previous experiences, $m \in M$, constructed by $\boldsymbol{LTM}$ to produce a novelty state, $n \in N$:

$$\boldsymbol{N} : C \times M \to N \tag{3.19}$$

Both novelty and surprise can be detected by comparing the current conceptual state of an agent with memories of previous experiences.

3.2.1.1 Novelty Detection

A novelty detector computes the novelty of a situation on the basis of all of the previous experiences of the agent. The determination of novelty assumes that future situations are likely to be similar to previous situations. Depending on the nature of the memory implemented in the agent, the determination of novelty may also assume that a situation in the near future is likely to be similar to one that has just past. To detect novelty an agent must:

1) Construct a categorisation of the current situation.
2) Determine the probability of the categorisation.
3) Compute novelty as the inverse of the probability.

The conceptual process, $\boldsymbol{C}$, at time t uses $\boldsymbol{LTM}$ to retrieve a generalised representation of similar situations that stands for the situation's category. The task of the agent's novelty detector, $\boldsymbol{N}$, is therefore to determine the probability of a situation's category being constructed.

3.2.1.2 Surprise Detection

A surprise detector computes the surprise of a situation compared to the expected situation based on previous experience. The process for detecting surprises is slightly more involved that for detecting novelty because it involves a comparison across time:

1) Construct expectations of a future situation.
2) Compare the previously constructed expectations with perceptions.
3) Compute surprise as the difference between expectations and perceptions.

At time t, the conceptual processes, $\boldsymbol{C}$, of an agent may be used to construct expectations of a future experience as an element of c_t that are stored in STM for later reference. The conceptual process may construct expectations either by getting $\boldsymbol{LTM}$ to provide a memory of a similar experience, m_s, where $0 \leq s \leq t$. or by reasoning about the consequences of taking an action, a_t.

At a later time u, where $u > t$, (e.g. after the agent has performed a_t) the agent's novelty detector, $\boldsymbol{N}$, can calculate an expectation-based measure of novelty by retrieving the previously constructed expectations in the conceptual state, C_t and compare them with the current conceptual state C_u to determine the novelty of the situation N_u as the difference between C_t and C_u.

3.2.2 Interestingness Function

The interestingness of a situation, as discussed in Section 2.4, is a measure of the importance of the situation with respect to an agent's existing knowledge; interesting situations are neither too similar nor too different from ones previously experienced. The interestingness function determines a value for the interestingness of a situation, $i \in I$, based on the novelty detected, $n \in N$:

$$\boldsymbol{I} : N \to I \qquad (3.20)$$

3.2.3 Curiosity

Curiosity is a meta-level conceptual process that monitors the conceptual process, $\boldsymbol{C}$, to estimate the potential of future states for learning. Curious conception is denoted in the following agents as a function, $\boldsymbol{C}$, with a similar signature as the conceptual process for a non-curious agent, $\boldsymbol{C}$, except that it monitors the conceptual state of the agent and not the perceptual state, and that it produces a curious state, $x \in X$, containing goals:

$$\boldsymbol{X} : C \times M \times STM \times I \to X \qquad (3.21)$$

The conceptual function $\boldsymbol{C}$ is updated to take the curious state of the agent into account when conceiving of the situation:

$$\boldsymbol{C} : M \times X \times STM \to C \qquad (3.22)$$

Figure 3.4 illustrates the architecture of a curious agent: it shows the communication between the conceptual function $\boldsymbol{C}$ and the curiosity function $\boldsymbol{X}$. The purpose of the communication between the conceptual and curiosity processes is to determine new goals based on interestingness. The interestingness of a situation for a curious agent is based on the novelty detected. Therefore $\boldsymbol{X}$ communicates with two other processes, $\boldsymbol{N}$ and $\boldsymbol{I}$, that determine the novelty and interestingness of a situation respectively.

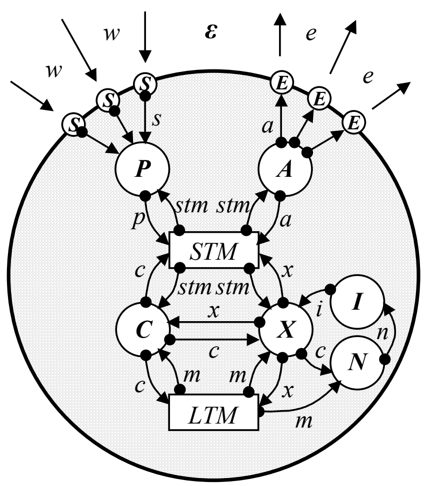


Figure 3.4: A curious agent.

3.3 MULTIPLE AGENTS

This section presents a framework for developing multi-agent models of social creativity by adapting Liu's dual generate-and-test model to use curious design agents. This approach to modelling social creativity is called artificial creativity.

3.3.1 Liu's Dual Generate-and-Test Model of Creativity

Recognising the need for a unified model of creativity in design computing, Liu (2000) presented a synthesis of the personal and socio-cultural views of creativity in a single model. Liu realised that the existing models of personal creativity complemented the socio-cultural models by providing details about the inner workings of the creative individual missing from the models of the larger creative system.

Liu proposed a dual generate-and-test model of creativity as a synthesis of Simon et al's model of creative thinking and Csikszentmihalyi's systems view. As its name suggests, the dual generate-and-test model of creativity encapsulates two generate-and-test loops: one at the level of the individual and the other at the level of society. The generate-and-test loop at the individual level, illustrated in Figure 3.5(a), provides a model of creative thinking, incorporating problem finding, solution generation and creativity evaluation. The socio-cultural generate-and-test loop models the interactions among the elements of Csikszentmihalyi's systems view of creativity, as illustrated in Figure 3.5(b). In particular, it captures the role that the field plays as a socio-cultural creativity test; ensuring that works that enter into the domain are considered creative by more that just its creator. In Liu's model, the domain has a rather passive role as a supplier of starting points for new generate-and-test cycles — a more dynamic model of the domain is discussed in Section 6.2.5.4 as the subject of possible future research. The combined dual generate-and-test model of creativity is illustrated in Figure 3.5(c).

Liu's model unifies Simon et al's and Csikszentmihalyi's models of creativity to form a computational model of creativity that shows how personal and socio-cultural views of creativity can be modelled in a single system. Compared to Boden's model of creativity, the dual generate-and-test model of creativity models both the P-creativity and H-creativity of individuals using the generate-and-test loops at different levels. Using the language of Gardner we may say that what distinguishes small-c creativity from big-c creativity is that big-c creativity affects changes to the domain whereas small-c creativity does not.

Liu's dual generate-and-test model shows that it is possible to cast Csikszentmihalyi's systems model in computational terms and thereby provides us with a useful basis for a framework for developing models of artificial creativity. Before developing Liu's model further, we will examine the requirements of a computational model of artificial creativity.

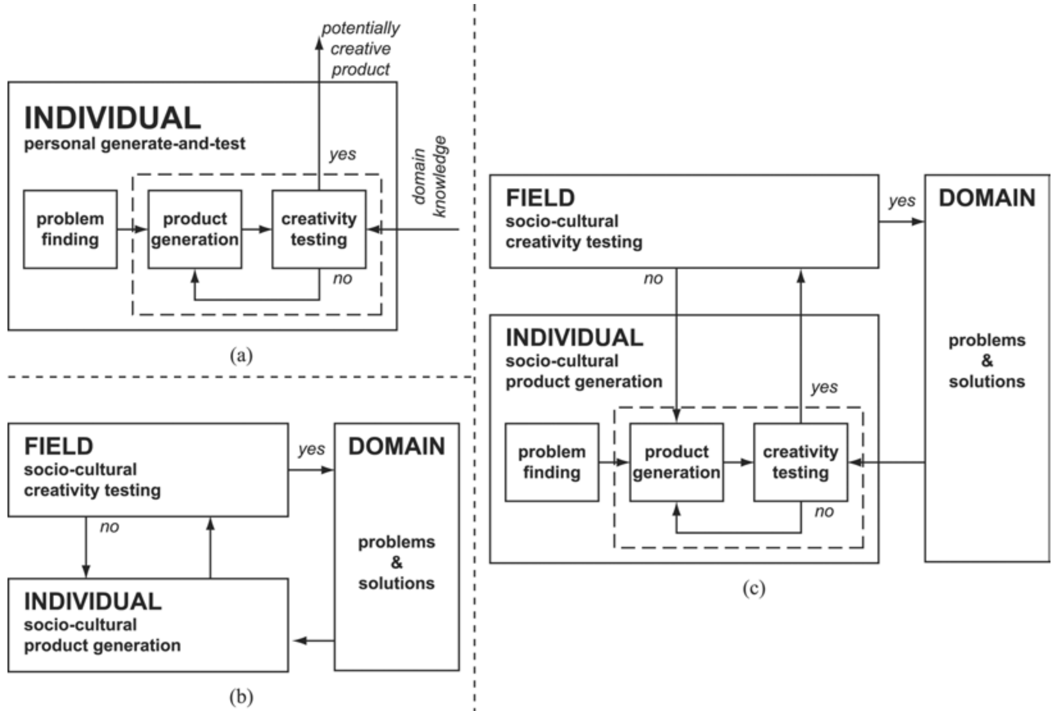


Figure 3.5: Liu's Dual Generate-and-Test Model of Creative Design: (a) the personal generate-and-test model, (b) the socio-cultural generate-and-test model, (c) the combined dual generate-and-test model.

3.3.2 Artificial Creativity

The artificial creativity approach that is presented here is similar to Langton's approach to developing computational models of Artificial Life (Langton, 1989). The essential requirements of a computational model of artificial creativity are:

- The model contains a society of agents situated in a cultural environment.
- There is no agent that can direct the behaviour of all of the other agents.
- There are no rules in the agents or the environment that dictate global behaviour.
- Agents interact with other agents to exchange artefacts and evaluations.
- Agents interact with the environment to access cultural symbols.
- Agents evaluate the creativity of artefacts and other agents.

Many of the requirements of a computational model of artificial creativity are similar to the requirements of a computational model of Artificial Life. Although some of the details are different, both types of models consist of a population of agents, and both require that there are no rules or agents that can dictate global behaviour. An additional requirement of artificial creativity agents not found in the requirements of Artificial Life is that the agents in an artificial creativity model must be able to make evaluative judgements about the creativity of products in order to implement the personal and socio-cultural creativity tests found in Liu's model. This ability of curious design agents to make these judgements is fundamental to the implementation of artificial creativity systems.

To illustrate the approach, consider how one would model a society of artists. First, we would define a repertoire of behaviours for different artistic agents and create lots of these agents. We would then start a simulation run by specifying some initial social configuration of the agents within a simulated cultural environment. From this point onwards the behaviour of the system would depend entirely on the interactions between different agents and the interactions between the agents and their cultural environment. Importantly, there would be no single agent that could enforce a definition of creativity by controlling the behaviour of all of the other agents. In addition, there would be no rules in the agents or in the environment that would define a global definition of creativity. The notions of whom and what are creative held by the society would emerge from the multiple notions of creativity held by the individual agents.

3.3.3 The Importance of Emergence

The requirements of artificial creativity are designed to model the emergence of phenomena in societies of agents consistent with creativity in human societies. Emergence is an important feature of artificial creativity systems, where the behaviour at a certain level of the creative system arises from interactions at lower levels. Cariani (1991) distinguished three types of emergence: computational emergence, thermodynamic emergence and emergence-relative-to-a-model.

Computational emergence is most often used to describe work in artificial life research. Complex global structures or behaviours arise from local computational interactions. In artificial life research, the stable patterns in cellular automata, and the flocking behaviour of simulated birds are examples of emergent phenomena. Cariani argues that from the perspective of a programmer with complete access to the computational elements of an artificial life simulation there is nothing emergent in them because at some level it must be deterministic in order to run on a computer. However, for an observer with an incomplete model of the computation, artificial life can provide important insights, as Cariani writes:

> "The interesting emergent events that involve artificial life simulations reside not in the simulations themselves, but in the ways that they change the way we think and interact with the world."

Thermodynamic emergence can be characterized as the emergence of order from noise. Stochastic processes at a micro-level form discrete macro-level structures or behaviours. In physical systems, temperature and pressure are examples of thermodynamic emergence. Temperature and pressure are emergent properties of large ensembles of molecules and are due to interactions at the molecular level. An individual molecule possesses neither temperature nor pressure; they are properties that only emerge when many molecules are brought together.

Emergence-relative-to-a-model is defined by Cariani as "deviation of the behaviour of a physical system from an observer's model of it." Emphasizing the requirement to have a model by which to judge the emergence within a system under observation. Importantly, while an artificial life simulation does not support emergence-relative-to-a-model for an omniscient programmer (at least within the confines of the simulation) such simulations can support emergence-relative-to-a-model for an artificial agent within the closed world with incomplete knowledge of the whole system.

In artificial creativity, the socio-cultural evaluations of whom and what are creative are emergent phenomena in the emergent-relative-to-a-model sense described above; no individual can dictate the collective evaluations of whom and what are creative, they can only try to influence other individuals by exposing them to their products and their personal evaluations. The emergence of macro-level creativity from the interactions of individuals at the micro-level is illustrated in Figure 3.6.

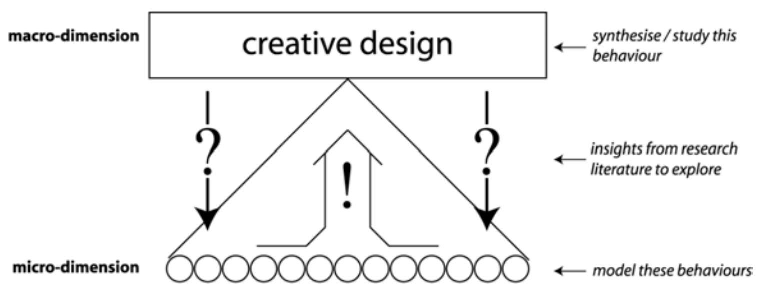


Figure 3.6: A behaviour-based approach to the study of emergent creative behaviour at the level of society by modelling the behaviour of individuals (after Langton, 1989).

In Boden’s terms we might be tempted to say that H-creativity is emergent whereas P-creativity is not because the processes that implement P-creativity test are fixed. However, in the artificial creativity system described later the interaction between agents and the continual learning of the agents through exposure to new artefacts mean that what an agent considers to be P-creative is an emergent property of the whole system. An individual embedded within an artificial creativity system is affected by its socio-cultural context such that it will not produce the same P-creative products as it would in isolation. Hence, both H-creativity and P-creativity must be considered emergent properties of creative systems.

3.3.4 Adapting Liu’s Model to Artificial Creativity

A critical aspect of Liu’s model that must be addressed to develop computational models of artificial creativity is the definition of the socio-cultural creativity test. A literal implementation of Liu’s model would produce a separate process that would model the socio-cultural creativity test. This is a viable solution for modelling some aspects of creativity, as demonstrated by the computational model developed by Gabora to study the memetic spread of innovations through a simulated culture Gabora (1995; 1996; 1997). Colton (2000) applied a similar socio-cultural creativity test to assess the increase in creativity due to the co-operation of agents searching a space of mathematical possibilities using different search heuristics. However, implementing a single function, or agent, that model a socio-cultural creativity test would violate one of the requirements for artificial creativity outlined previously, i.e. that no rule or agent should direct global behaviour.

Liu does not go in to details about the definition of this function but it appears that he considers this function to be outside the scope of computational models and something that can only be implemented by some form of interaction with human society. Many computational models developed reinforce this view by concentrating on the constrained generation of novel ideas in their computational models and relying on users to evaluate the creative worth of these ideas. For example, see Clancey (1997) for a discussion of the social situatedness of Harold Cohen's AARON.

To computationally model the behaviour of creative societies, it is necessary to define a socio-cultural creativity test without violating the requirements of artificial creativity. The key to solving this problem is to realise that the personal creativity test inside each individual can be used to develop a socio-cultural test for creativity. The socio-cultural creativity test can be modelled by permitting the communication of artefacts and evaluations of personal creativity between individuals. An illustration of two individuals communicating creativity evaluations is illustrated in Figure 3.7.

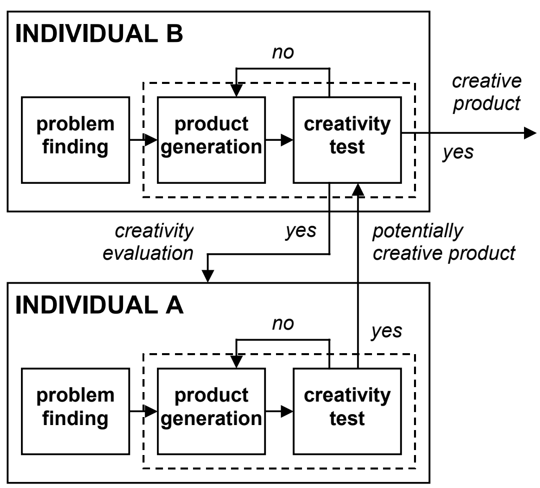


Figure 3.7: The communication of evaluations between individuals and its integration into the individual generate-and-test cycle.

In the interaction illustrated in Figure 3.7, Agent A communicates an artefact that it considers to be creative, i.e. that passes its personal creativity test, to Agent B. Agent B evaluates the artefact according to its own personal creativity test and sends its evaluation back to Agent A. In this way, Agent B can affect the generation of future artefacts by Agent A by rewarding Agent A when it generates artefacts that Agent B considers to be creative. More subtly, Agent A can affect the personal creativity test of Agent B by exposing it to artefacts that Agent A considers to be creative, because the evaluation of creativity involves an evaluation of novelty, Agent A affects a change in Agent B's notion of creativity by reducing the novelty of the type of artefacts that it communicates. By exposing Agent B to artefacts that Agent A considers to be creative, because they are novel and yet understandable, it can alter the evaluation of creativity made by Agent B.

Agent-centric evaluations of creativity permit the emergence of socio-cultural definitions of creativity as the collective function of many individual evaluations. Without agent-centric evaluations of interestingness the collection of agents would simply represent parallel searches of the same design space. To implement the socio-cultural creativity test as a collective function of individual creativity tests a communication policy is needed. A simple communication policy would be for agents to communicate a product when their evaluation of that product is greater than some fixed threshold. More complex communication policies might incorporate more strategic knowledge about when to communicate and who to communicate with.

To complete the implementation of the field as a collection of individuals, the individuals must be given the ability to interact with the domain according to some domain interaction policy. A simple domain interaction policy would follow the communication policy above and allow agents to add products of the generative process if the personal creativity evaluation is greater than a domain interaction threshold. This approach is illustrated in Figure 3.7. However, to ensure some level of social agreement before the addition of products to the domain, a slightly more complex domain interaction policy ensures that no individual is allowed to submit their own work to the domain. Thus, at least one other agent must find an individual's work creative before it is entered into the domain.

Making these amendments to Liu's dual generate-and-test results in the model of socio-cultural creativity illustrated in Figure 3.8.

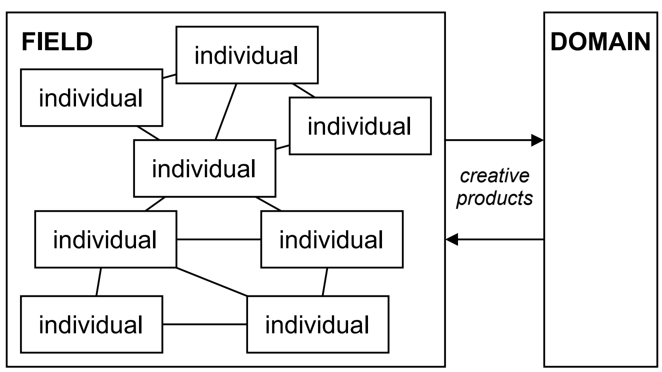


Figure 3.8: The artificial creativity model of socio-cultural creativity.

3.4 CONCLUSIONS

The frameworks for curious design agents and artificial creativity presented in this chapter show that:

1) A small number of additional functions are required to transform an agent developed using a conventional framework to one that models creativity.

2) Using an approach familiar from the study of artificial life, curious design agents can be used to model socially situated creative behaviour.

Modelling curiosity requires the addition of only a few functions that monitor the conceptual state of the standard agent. The addition of functions to detect novelty, determine its interestingness and takes actions to promote future interesting experiences, provides curious design agents with the sort of autonomy that previous models of creativity have lacked.

The autonomy of curious design agents for determining what is interesting, and therefore potentially creative, is the key to adapting Liu's dual generate-and-test model of designing. The artificial creativity approach substitutes the monolithic social test of creativity found in Liu's model with a distributed agreement that emerges from the communication of individuals.

Chapter 4

Implementing Curiosity

At the heart of a curious design agent are the core memory units ***STM*** and ***LTM***. One of the most critical decisions when implementing curious design agents is how to implement these components. Different implementations of short-term and long-term memory will produce different curious behaviour because curiosity is based upon the ability of these implementations to learn the design space that the agent is exploring.

This chapter examines some of the options available to curious design agent developers in terms of the technologies used to build components and the resulting differences in behaviour of the agents.

The first half of this chapter provides some details of the technologies used in this research to implement functions of short-term and long-term memory, novelty detection, interest evaluation and curious goal setting. In the case of long-term memory, attention is given to the learning behaviours of the different implementations as these have a significant effect on the behaviour of agents developed using them. The second half of this chapter gives some examples of curious behaviour in a simple environment that is presented as a spatial analogue to a design space. The differences that result from using different implementations of the various components are illustrated in this visual environment to give a feel for the types of behaviours that can be expected in the more complicated design applications that follow.

The implementation of processes for sensing, perceiving, acting and effecting is domain specific and is dealt with in the sections covering applications of curious design agents. Concept formation, learning and curiosity are more general and the implementations of these functions are discussed here.

4.1 SHORT-TERM MEMORY

Short-term memory can be implemented simply as a store of recent variables produced by the various processes that interact with it. In this way, short-term memory is modelled as an accurate record of recent experiences that go to make up a situation. Of course, more complex models of short-term memory are possible, but a simple store of important variables has sufficed for the agents presented in the following chapters.

Simple agents have only a small number of experiences that need to be stored in ***STM*** at a time but more complicated agents may require limits to be placed on the number of experiences that ***STM*** can hold and hence some sort of decision criteria to determine which memories are important enough to store in ***STM*** (see Waltz (1998) for a discussion of the situated nature of importance.)

4.2 LONG-TERM MEMORY

Implementing the construction of memories by ***LTM*** is a more complex problem than implementing short-term memory. To implement long-term memory a process is required that can store representations of previous experiences and their associations with other experiences such that they can be retrieved at a later time with a new experience may be only a partial match to the original. This type of memory is sometimes called an *associative memory*. As their name suggests associative memories store associations between stored representations. To store an association between two representations they are presented simultaneously to the associative memory and it learns a mapping from one to the other. In an agent, an associative memory can be used to map between different features of an experience such that expectations of other features, either in the same experience or in future ones, can be predicted.

When an associative memory is given the same representation twice, as both input and output of the mapping, it is often called an *auto-associative memory*. In an agent, an auto-associative memory can be used to recall a previously stored representation of a given experience. This may not seem very useful, if the memory were perfect then the output of the memory would be the same as the input. However, as discussed above, long-term memory is necessarily a constructive process because of the prohibitive requirements on storage capacity and recollection time of a perfect long-term memory implemented as a database that stored every experience perfectly. Instead, ***LTM*** constructs memories using a process that attempts to store the most important aspects of an experience by removing redundancies. Importantly for an agent developer, the processes that auto-associative memories employ to achieve the required compression can have a significant effect on the experiences that are recalled in response to a situation. Instead of recalling perfect memories of previous experiences an auto-associative memory will recall a generalised representation of an amalgam of similar experiences that can be used to compare an experience against a history of previous similar experiences. This is the basis of novelty detection in many of the agents described in the remainder of this thesis.

4.2.1 Neural Networks

Several different types of neural networks have been used to implement the learning systems in curious design agents. Self-Organising Maps (Kohonen, 1993; 1995) and networks based on Adaptive Resonance Theory (Carpenter and Grossberg, 1987a; 1987b; 1990; Carpenter et al., 1991a; 1991b; 1991c; 1992) have proved to be useful and complementary technologies for implementing unsupervised learning. The algorithms for these two different approaches to unsupervised learning are described briefly in this section, because the dynamics of the resulting networks play important roles in the behaviour of curious design agents. Self-Organising Maps and Adaptive Resonance Theory networks have quite different behaviours when learning conceptual spaces, which make them good for different design tasks depending on the characteristics of the design space and the novelty to be detected.

4.2.2 Self-Organising Maps

Self-Organising Maps (SOMs), also known as the Self-Organising Feature Maps or Kohonen Networks, are some of the most popular neural network models. A self-organising map consists of a lattice of neurons that are used to represent different categories of inputs (Kohonen, 1993). Each neuron has an associated vector of weights of the same dimension as the inputs. When a new input is presented to the SOM each neuron compares the similarity of its weight vector to the inputs. The neuron with the best matching weights is declared the winner. Learning is accomplished by updating the winner to reduce the difference between its weights and the inputs. In addition, the neurons within a neighbourhood around the winner are updated to reduce the difference between their weights and the inputs. This process results in a topographic map of the input space, with similar categories being represented by nearby neurons.

Self-Organising Maps have proved to be useful in many applications (see Kaski, Kangas and Kohonen, 1998). The unsupervised nature of the SOM learning process means that little domain specific knowledge is required about the characteristics of input data to apply a SOM to a domain. For this reason SOMs have been used to visualise data in complex domains where little knowledge exists to discover the inherent categories. SOMs have also been used as detect features of a range of inputs, rather than to classes of whole inputs; hence they have sometimes been called Self-Organising Feature Maps.

4.2.3 Adaptive Resonance Theory Networks

Grossberg (1976) introduced the Adaptive Resonance Theory (ART) to provide a framework for investigating how people can rapidly and stably learn in real-time about an ever-changing world. As a parallel development to the Adaptive Resonance Theory a series of neural network architectures have been developed with increasingly powerful learning, pattern recognition and hypothesis testing capabilities (Carpenter and Grossberg; 1987a; Carpenter and Grossberg; 1987b; Carpenter and Grossberg; 1990; Carpenter et al.; 1991; Bartfai; 1994).

The central result of the development of the ART-based networks is the solution to the stability-plasticity dilemma (Carpenter and Grossberg, 1987a). The stability-plasticity dilemma states that an adequate learning system must be flexible enough to generate recognition codes while remaining stable enough to guard against relentless recoding caused by irrelevant inputs. The solution to the stability-plasticity dilemma provided by ART networks is to separate the problem into two parts by distinguishing between familiar and unfamiliar input patterns using a measure of confidence in category matches. The familiarity of an input is determined by an orienting subsystem that maintains a measure of *tolerance* for new patterns. If the difference between an input and the prototypes represented by the neurons of the network is greater than the tolerance value then the input is considered unfamiliar.

Unlike SOM networks, the number of neurons in an ART network is variable. Familiar patterns are handled in much the same way as inputs are handled by neurons in a SOM and the weights of the neuron are updated to better represent the input. When unfamiliar input patterns are encountered a series of reset signals sent by the orienting subsystem of the network, eventually causes a new neuron to be added to the network, ensuring that previously learned categories are retained.

Baraldi and Alpaydin introduced the SIMPLIFIED ART algorithm that improves upon the efficiency of previous ART networks and simplifies the implementation without sacrificing the desirable qualities of stability and plasticity (Baraldi and Alpaydin, 1998). The ART networks used throughout this research were all implemented using the SIMPLIFIED ART algorithm.

4.2.3.1 Supervised ART Networks

The adaptive resonance theory has also been used to develop a supervised learning system called ARTMAP. The ARTMAP network builds upon the strengths of the ART network by creating a means of linking hypothesis testing with the reorganization of knowledge (Carpenter and Grossberg, 1991). An ARTMAP network consists of two ART networks connected by an associative MAP FIELD with an orienting subsystem.

The architecture for ARTMAP networks was developed as a biologically plausible model of associative learning in the brain. Weenink (1997) developed a simpler network ART-based architecture for supervised learning called CATEGORY ART that performs similarly to ARTMAP but is computationally more efficient and easier to implement. The curious agent presented below uses CATEGORY ART networks to detect novelty.

4.2.4 Learning Behaviour

Many of the behaviours of curious agents can be traced back to the learning behaviours of the neural networks that are used to implement long-term memory so it is worthwhile spending a little time examining the learning behaviours of these networks.

4.2.4.1 Self-Organising Maps

Figure 4.1 illustrates the behaviour of a 10x10 SOM as it learns a two-dimensional space. It shows that SOM slowly expands to fill the space and then adjusts the location of the neurons to fill the space evenly.

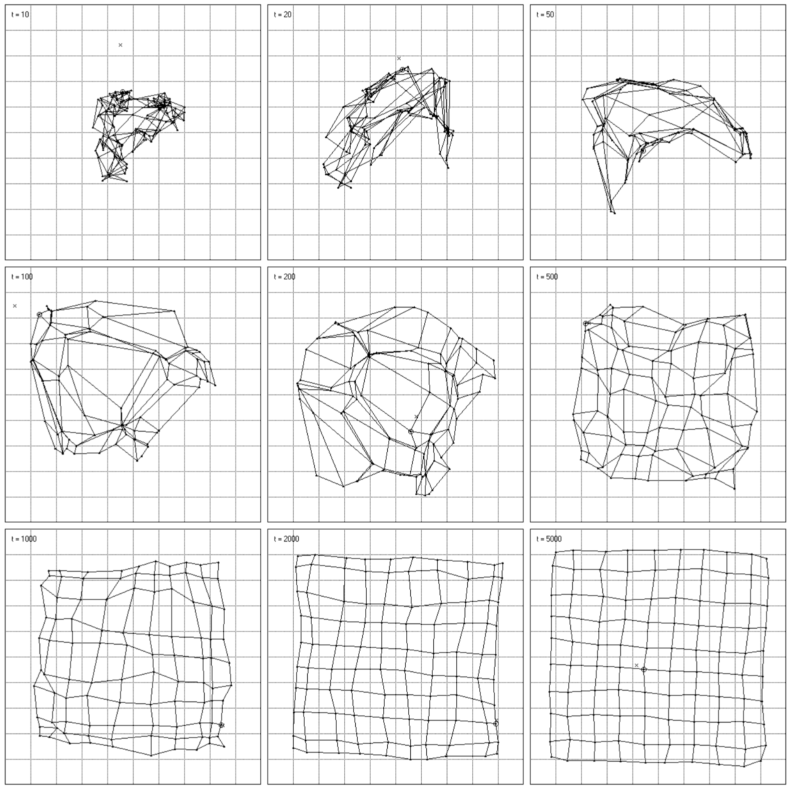
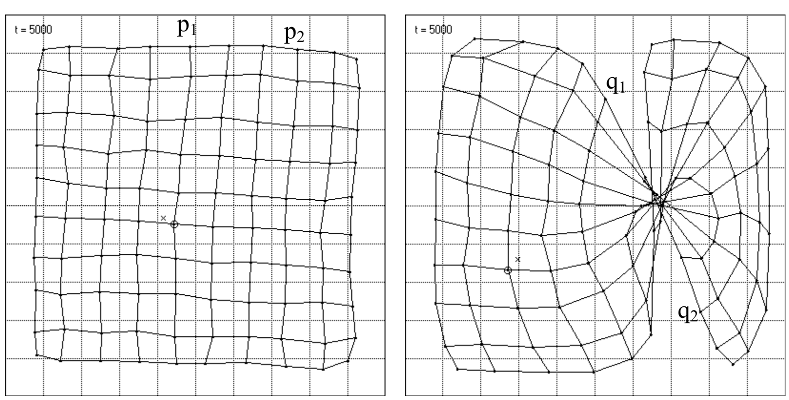


Figure 4.1: The mapping of a two-dimensional space using a self-organising map. Sequence shows inverse mapping of neurons at time t = 10, 20, 50, 100, 200, 500, 1000, 2000, 5000. The learning rate of the SOM was fixed at 0.1 with a rectangular neighbourhood of radius 3.

The SOM learning algorithm ensures that the maps produced are topology preserving within neighbourhoods, as determined by the neighbourhood function, but it cannot guarantee that the map will preserve the topology of the input space for the whole map. Figure 4.2 illustrates one of the most common artefacts of the learning process, a "twist" in the SOM that reduces the degree to which the map preserves the topology of the map on a larger scale than that of the neighbourhoods used to train it.



(a) (b)

Figure 4.2: Self-organising maps that have learned mappings for a two-dimensional space after 20000 time steps: (a) shows a good mapping, (b) shows a bad mapping.

Two neurons mapped onto the space at the positions marked in Figure 4.2a as p_1 and p_2, which are relatively close in the input space as they should be for a properly trained SOM. In contrast, Figure 4.2b shows the equivalent neurons have been mapped to positions q_1 and q_2, on opposite sides of the input space. Obviously, over this region the second map does not preserve the topology of the input space.

Kohonen (1995) recommends that to avoid such problematic mappings the neighbourhood function should initially cover more than half of the neurons to ensure an early large-scale ordering on the neurons and that the neighbourhood should decrease in size over time to refine the smaller-scale properties of the map. Unfortunately, such an approach is not generally useful in systems that must learn on-line, such as autonomous agents, as there may be no generally applicable method for determining when the large-scale structure of the input space has been learned sufficiently well to decrease the size of the neighbourhood function.

4.2.4.2 Simplified ART Networks

To compare the behaviour of ART networks with SOMs, an ART network was set the same task of learning a two-dimensional space as above. The behaviour of the ART network is illustrated in Figure 4.3.

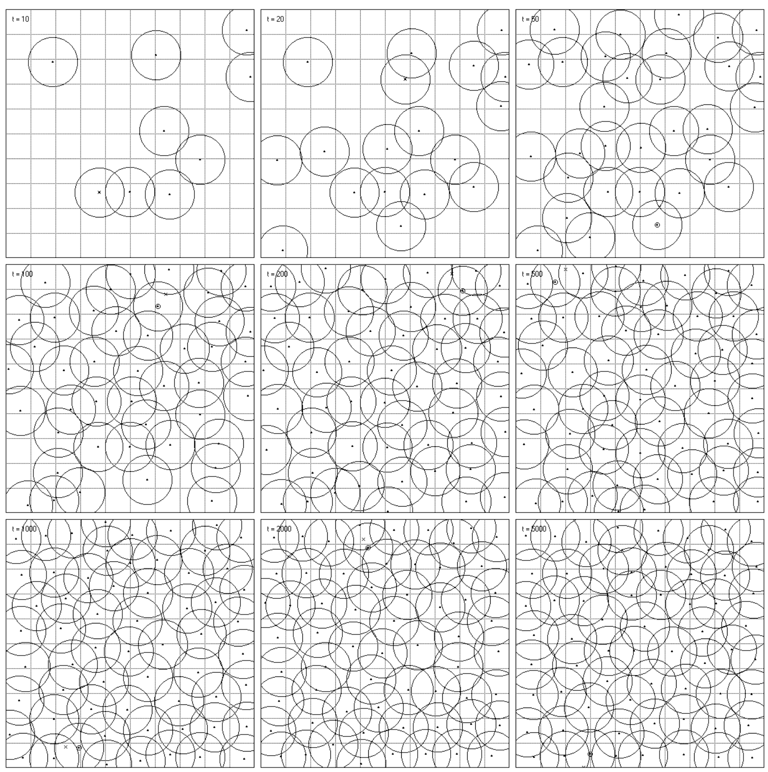


Figure 4.3: The mapping of a two-dimensional space using an ART network. Sequence shows inverse mapping of neurons at time t = 10, 20, 50, 100, 200, 500, 1000, 2000, 5000.

The ability of an ART network to refine its map of a space as required by adjusting the tolerance of neurons will become apparent when we look at the behaviour of ART networks designed for learning a mapping between domains.

4.2.4.3 Category ART Networks

Figure 4.4 illustrates the learning behaviour of a CATEGORY ART set the task of mapping a two-dimensional input space to a four-colour output space, where each of the four colours has been allocated to a different quadrant of the input space. The details of the colour categories are not important, only that they are allocated to the four quadrants of the input space.

Unlike the previous examples of ART networks Figure 4.4 shows that the CATEGORY ART network does not allocate the neurons evenly over the surface of the input space. Instead the CATEGORY ART network allocates many more neurons close to the boundaries of the four output categories to ensure that the boundaries are resolved sufficiently well.

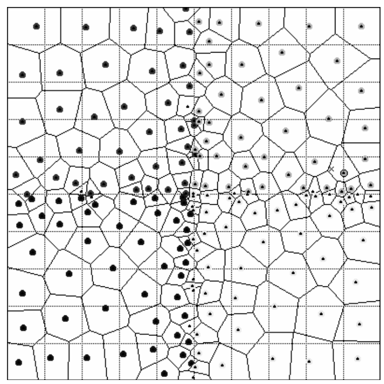


Figure 4.4: A CATEGORY ART network that has learned a mapping from a two-dimensional space to a four-colour space where the four colours have been assigned to the quadrants of the plane. The Voronoi cells of the neuron prototypes are shown.

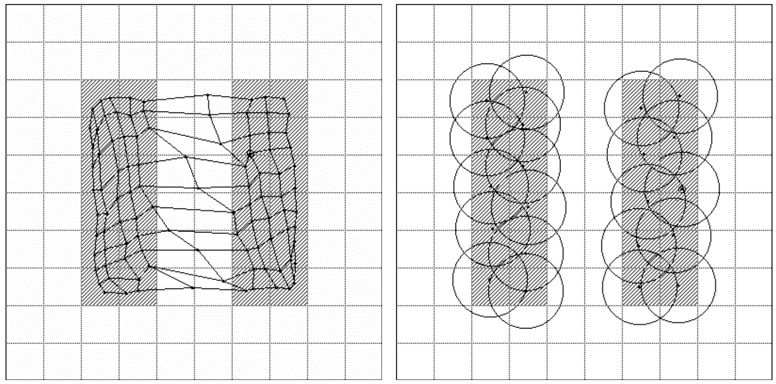
4.2.5 On-line Learning Behaviour

The differences in behaviour between SOM and ART networks become even more important when we consider that the order that inputs are given to these networks are not typically random in the agent-based applications.

4.2.5.1 Predicting Unseen Situations

An important difference between SOMs and ART networks is their ability to predict unseen situations on the basis of the inputs used to train them. The topology preserving qualities of SOMs make them ideal for tasks where the design space is continuous and it is desirable for ***LTM*** to make predictions of unseen situations that lay within the convex hull of previously experienced situations.

Figure 4.5 illustrates the behaviour of a SOM and an ART network given the task of learning categorisations for a two-dimensional space where all of the inputs come from two regions separated by a small gap. The regions from which the training inputs are taken are marked as the rectangular shaded areas. Figure 4.5 shows that the SOM allocates neurons to the space between the input regions but the ART network does not.



(a) (b)

Figure 4.5: An illustration of how a SOM allocates neurons to unseen situations while an ART network does not: (a) a SOM and (b) an ART network trained on the same disconitunous space of samples, indicated by the shaded region.

The mapping of neurons in the design space over unseen portions by SOMs make them a good choice for detecting truly novelty areas of the design space because a SOM will be able to generate reasonable expectations of the sorts of experiences that can be expected in the unseen region.

ART networks are more conservative in their predictions, providing confident predictions of unseen experiences only within the bounds of the tolerances of neurons close to the edges of each region.

4.2.5.2 Dragging Networks

A potential problem arises with the use of SOM and ART networks that is a consequence of the order that inputs are presented because both learning algorithms are sensitive to the order of presentation.

The agents used in this research present similar inputs to their neural networks in close succession along paths through the input space. Figure 4.6 shows how a SOM and an ART network, initially trained on a central region, respond to a sequence of inputs taken along a "random walk" to the top-right hand corner of the space. The SOM is dragged towards the corner as the inputs are presented. In contrast, the ART network adds new neurons to account for the new inputs along a similar random path.

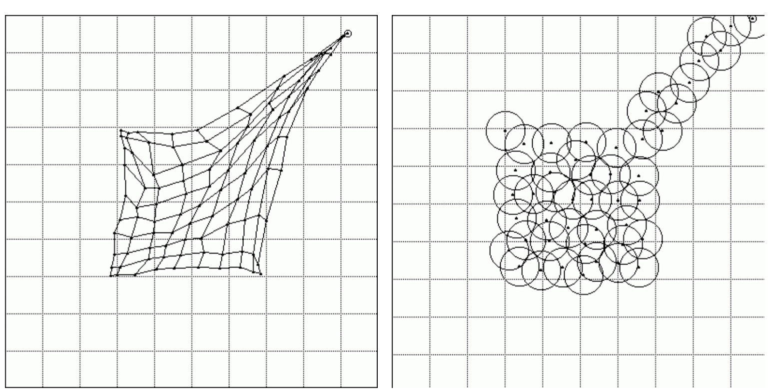


Figure 4.6: A SOM and an ART network trained on central region and then exposed to a series of inputs along a random walk to the top right corner: (a) shows how the SOM is deformed as the corner is "dragged" and (b) shows how the ART assigns new neurons.

According to this result SOMs are more susceptible to deformation as a consequence of the order that inputs are presented. This might be the expected result given that ART networks were designed to resolve the stability-plasticity dilemma, however, Figure 4.7 shows a possible problem with ART-based learning. If the distance between successive inputs is less than the distance moved by the neuron as a result of learning, a single neuron can be separated from all of the others and slowly dragged around the input space.

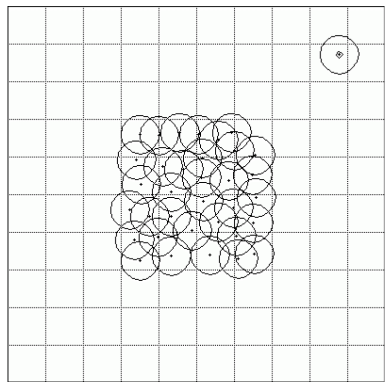


Figure 4.7: An ART network initially trained on randomly chosen points in the centre of the space and then exposed to a series of points along a slow random walk to the top right corner. Shows how a single neuron has been dragged along the path.

These different behaviours affect the determination of novelty and have to be borne in mind when the behaviour of curious design agents is being considered.

4.2.6 Novelty Detectors

Novelty based on the atypicality of an input is generally calculated based on some categorisation error function. While detecting novelty based on infrequency requires that the input be confidently recognised before its frequency can be estimated. For SOM and ART networks, a simple error measure is the Euclidean distance between the closest category prototype and the input pattern. Novelty based on this error measure can determine the atypicality of the input.

To recognise the novelty of infrequent or uncommon patterns, Marsland et al. (2000c) have proposed the HABITUATED SOM (HSOM) architecture. In an HSOM each neuron of the SOM is connected to an output neuron via a habituating synapse. The habituating synapse reduces the efficacy with which it transfers activation from the map neuron to the output neuron with use according to Stanley's model of habituation (Stanley, 1976). At time t, the synaptic efficacy, y, decreases according to the following equation (from Marsland, 2000c):

$$\tau \frac{dy}{dt} = \alpha[y_0 - y] - S \tag{4.1}$$

where y_0 is the original value of y, τ and α are constants governing the rate of habituation and recovery respectively, and S is the stimulus presented. In an HSOM, S is the output of a map neuron. Novelty is calculated by the output neuron attached to each of the map neurons as the sum the map neuron activations attenuated by the habituated synapses.

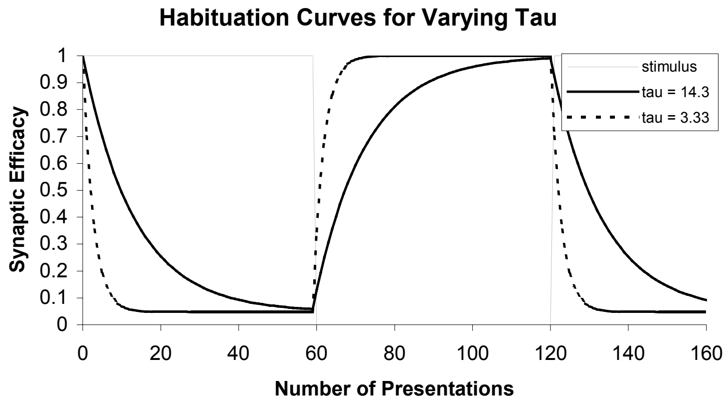


Figure 4.8: The effects of habituation on synaptic efficacy for different values of τ. In both cases $\alpha = 1.05$. The stimulus curve indicates the output of the SOM neuron connected to the habituating synapse.

4.2.6.1 Combining Measures of Novelty

Different measures of novelty may need to be combined to produce a measure of the novelty for a situation. The simplest way to combine novelty evaluations is to treat them as if they were independent estimates of the probability of an aspect of the situation being novel. The probability of the situation, taken as a whole, being novel can therefore be calculated as the product of the different novelty measures.

4.2.7 Interest and Boredom

The models of interest and boredom implemented in curious design agents have a big impact on the behaviour of the agents. The two models used in this research use a linear hedonic function and a hedonic function based on the Wundt curve (discussed in Section 2.4.2).

The linear hedonic function has been explicitly used by previous researchers (e.g. Schmidhuber, 1997) or has been implicitly assumed by others who have not included interest as distinct from novelty (e.g. Marsland et al., 2000). A linear hedonic function can be thought of as an approximation to the Wundt Curve close to the origin where it is safe to seek as much novelty as possible. A linear hedonic function is appropriate when only a small amount of novelty is expected during the lifetime. In many cases extreme novelty is either of little worth because it cannot be learned, in these cases a hedonic function based on the Wundt Curve will be preferable because it models an aversion to extreme novelty.

4.2.7.1 Modelling Interest using a Linear Hedonic Function

A simple way to model interest is to linearly scale the novelty detected (Schmidhuber (1997). The interest at time t, i_t, is equal to the linear reward, $\boldsymbol{R}(n_t)$.

$$\boldsymbol{R}(n_t) = \rho n_t + R_0 \qquad (4.2)$$

Where ρ is the slope of the reward function and R_0 is the base reward for $n_t = 0$. Typically $R_0 = 0$ and ρ is selected so that the maximum value of $\boldsymbol{R}(n_t) = 1$, i.e. $\rho = 1 / \max(n_t)$, where $\max(n_t)$ is the maximum value of $n_t \ \forall\ t$.

4.2.7.2 Modelling Interestingness using the Wundt Curve

The Wundt curve can be approximated as the difference of a reward function $\boldsymbol{R}(n_t)$ and a punishment function $\boldsymbol{P}(n_t)$:

$$i_t = \boldsymbol{R}(n_t) - \boldsymbol{P}(n_t)$$

$$\boldsymbol{R}(n_t) = \frac{R_{\max}}{1+e^{-\rho_R(n_t - R_{\min})}}$$

$$\boldsymbol{P}(n_t) = \frac{P_{\max}}{1+e^{-\rho_P(n_t - P_{\min})}} \qquad (4.3)$$

Where $R_{\max}$ is the maximum reward, P_{max} is the maximum punishment, ρ_R and ρ_P are the slopes of the reward and punishment sigmoid functions, $R_{\min}$ is the minimum novelty to be rewarded and $P_{\min}$ is the minimum novelty to be punished.

4.2.7.3 Modelling Boredom

Boredom is an internal state of a curious agent that tracks the interestingness of situations over time. A lack of interestingness will make a single situation "boring" but it requires a sustained lack of interesting situations for an agent to become bored. To model boredom an agent must track the degree of its accumulated interest in successive situations. The current accumulated interest at time t, y_t, is a fraction, δ, of the accumulated interest of the agent at time t-1, y_{t-1}, plus the interest of the agent in the current situation, i_t.

$$y_t = \delta y_{t-1} + i_t \ \text{ where } \ 0 \le \delta \le 1 \qquad (4.\ 4)$$

A state of boredom can be declared for an agent when its arousal falls below a "boredom threshold", i.e. when $y_t < \beta$, where β is the boredom threshold.

4.2.8 Curiosity

Curiosity is the outward behaviour that an agent exhibits and as such it requires that the agent take some action. A simple heuristic for navigating design spaces uses the detected novelty to determine how much design variables should be changed, i.e. if the current design is interesting generate a similar design as it is also likely to be interesting else if the current design is not interesting generate a dissimilar design to begin a search for interesting designs elsewhere in the design space.

4.3 CONCLUSIONS

This chapter has provided some important background information for anyone that wants to implement curious design agents:

1) Appropriate technologies have been identified for implementing models of curiosity, e.g. self-organising artificial neural networks.
2) Some of the potential consequences of choosing one technology over another for the purposes of implementing curiosity have been explored.

This chapter has shown that curiosity is relatively easy to implement, it can be implemented using standard and relatively simple components: in particular, neural networks, novelty detectors and simple reward functions. The following chapters examine some of the behaviours that emerge when curious design agents explore design domains.

Chapter 5

Curious Design Methods

The implementation described in the preceding chapter provides the core technology for developing curious design agents: applying a curious design agent to a new domain simply requires the addition of appropriate sensors, perceptors, actors and effectors. It has been possible to apply curious design agents to a number of different domains with relative ease and thereby explore the behaviour of curious agents in multiple domains using different approaches to designing.

The aim of this chapter is to show that curiosity is a general-purpose search heuristic useful for exploring design spaces using different design methods. This chapter presents three curious design agents exploring different visual domains using three different design methods to explore these spaces:

1) Direct manipulation
2) Parametric configuration
3) Design tool-use

The use of these methods in the following experiments is intended to show that the model of curiosity is generally applicable in models of designing and not restricted to a particular method of working. In particular, the use of different design methods shows that curious design agents can be applied at different levels of abstraction in the design process.

The three design methods used are discussed in the following section to provide a context for the experimental work that follows. The first curious design agent explores a space of rectilinear drawings using direct manipulation. The second curious design agent explores Spirograph patterns using parametric configuration. The third curious design agent explores a space of "genetic artworks" using an evolutionary design tool. Each application is presented by first examining motivations for exploring the particular space. Implementation details are given for the specific application of each agent, including details of application-specific processes of each agent. In each case, experimental results are followed by an application-specific discussion and some conclusions. A more general discussion of curious design agents is reserved for Chapter 7.

5.1 DESIGN METHODS

Design methods play an important role in determining what a designer can accomplish. Broadbent (1973) presents a survey of the different methods that designers, particularly architects have used to generate new designs. Design methods are often influenced by the development of technology, or the discovery of new insights into the nature of the world. The three classes of design methods presented here – direct manipulation, parametric design, and design tool use – are typical of the methods that designers engage in daily. The specific implementations – rectilinear sketching, pattern generation and interactive evolutionary design – are simple examples of these design methods but sufficient to show the potential of curious design agents.

5.1.1 Direct Manipulation

A digital computer, without the assistance of robotics, cannot directly manipulate anything other than the 0s and 1s that make up the binary data used to perform computations. Direct manipulation as used in this context means that an agent changes the same representation that it senses. When engaged in design by direct manipulation an agent affects the same binary data that it later senses. The data may be organised into a composite structure and the agent's effectors may manipulate these composite structures, however, the result is that the agent senses the same data that is affected. Direct manipulation permits the finest control over the design process and allows curiosity to play a role in every aspect of the design process from conceptual to detail.

5.1.2 Parametric Design

In contrast, an agent that engages in parametric design affects changes in some, presumably high-level, representation of a design artefact that in turn changes some other data held in the external environment that the agent later senses. There is at least one degree of separation between the effects of the agent on the environment and the data that is later sensed. When engaged in parametric design the agent uses a tool that renders an artefact according to supplied parameters. Parametric design allows an agent to explore domains where the skills required to produce an artefact using direct manipulation are not supported by its effectors or require skills that are too complex for the agent to learn.

5.1.3 Design Tool-Use

An agent that uses a design tool to search a design space does so by changing the process implemented by the tool and sensing the products of the new process. Unlike the tools used in parametric design, a design tool conducts an independent design process, possibly a complex one. Using a design tool allows an agent to take another step away from the specifics of the domain. The use of design tools allows simple design agents to explore complex design spaces without having to learn complicated design skills.

5.2 REFLECT-A-SKETCH

When designer's sketch they often discover unintended shapes emerging in their drawings. This phenomenon is commonly called 'shape emergence' and has been widely reported in the design research community, for a review of relevant studies see: Purcell and Gero (1998). Cognitive studies of designers suggest that the emergence of unintended shapes in a sketch plays an important role in the creative designing, especially in the early conceptual stages of designing.

5.2.1 Motivation

Reflect-a-sketch was developed as a computational model of reflection-in-action (Schön, 1983). More specifically, it attempts to model the reflective sketching processes observed in the activity of designers by Schön and Wiggins (1992) by modelling some 'different kinds of seeing'. In particular, Reflect-a-Sketch can recognise the unintended consequences of its moves. The development of Reflect-a-Sketch was motivated by the following conclusion made at the end of Schön and Wiggins' paper:

> When we think of designing … as a conversation with materials conducted in the medium of drawing and crucially dependent on seeing, we are bound to attend to processes that computers are unable — at least presently unable — to reproduce: the perception of figures or gestalts, the appreciation of qualities, the recognition of unintended consequences of moves. (Schön and Wiggins, 1992)

Reflect-a-Sketch was developed as a way of re-examining the current limitations of computers with respect to reflection-in-action. It was designed to produce simple rectilinear drawings, reflect upon those drawings, identify interesting (emergent) shapes and learn new drawing commands.

5.2.1.1 Computationally Modelling Shape Emergence

Computational models of shape emergence have previously been developed that can extract the implicit shapes within a drawing that are easily recognisable by human observers. Computational models of shape emergence have typically created an unstructured intermediate representation of a sketch and then identified emergent shapes by combining elements of the intermediate representation in new ways. Computational systems using infinite maximal lines (Gero and Yan, 1993) have proved successful in identifying emergent shapes (Damski and Gero, 1996), emergent shape semantics (Gero and Jun, 1995) and emergent patterns (Cha and Gero, 1998). Figure 5.1 illustrates a good example of the emergence of multiple shape representations from a single building floor plan using infinite maximal lines.

Alternative computational models of shape emergence have used bitmap images as intermediate representations. Image processing techniques are used to find emergent shapes by recognising structure in the bitmap representation. Liu (1993) used neural networks to identify previously learned emergent sub-shapes, Edmonds and Soufi (1992) used Gestalt operators to construct emergent groupings of similar shapes, and Tomlinson and Gero (1997) used a model of early visual processing developed by Grossberg and Mingolla (1985a, 1985b) to emerge optical illusions.

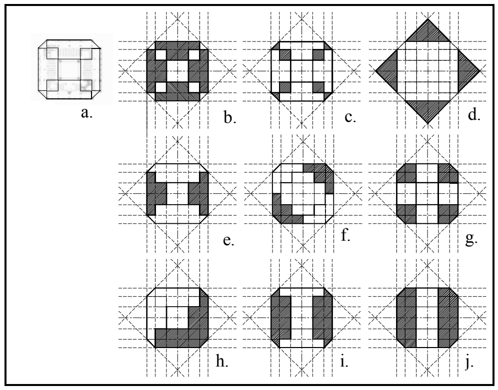


Figure 5.1: An example of the emergence of multiple representations for a floor plan design using infinite maximal lines (from Reffat and Gero, 1998).

To exploit emergence in future design tasks, designers must learn about the initially unintended consequences of their actions. Most of the computational models of shape emergence have lacked the ability to learn. As a consequence all of the emergent shapes discovered had to be considered "interesting" and presented to a user for further evaluation. In contrast, the computational model of shape emergence presented here is capable of learning to expect emergent shapes. During the sketching process, Reflect-a-Sketch focuses upon the novel and unexpected aspects of its sketch to learn new drawing skills and expand the repertoire of sketches that it can produce. An earlier version of Reflect-a-Sketch was presented in (Gero and Saunders, 2000).

5.2.2 Implementation

Schön and Wiggins (1992) describe the reflective sketching process in terms of 'moving' and 'seeing'. The architecture of a curious agent is described here in terms of two subsystems that implement the 'moving' and 'seeing' processes. The 'moving' subsystems include processes implementing sensing ***S***, effecting ***E***, perception ***P***, and action ***A***. The 'seeing' subsystems include a conceptual unit ***C*** and a curiosity module, ***X*** incorporating a novelty detector ***N*** and an interest function ***I***. The architecture of Reflect-a-Sketch is illustrated in Figure 5.2; the figure shows the representations constructed by each process.

5.2.2.1 Moving

The moving subsystem consists of the processes for drawing, sensing primitive elements, and perceiving more complex structures.

5.2.2.2 Drawing

Reflect-a-Sketch has very limited drawing and sensing capabilities: it can draw horizontal and vertical straight lines and it can sense the pixels values within a small window onto the canvas. Reflect-a-Sketch's external environment is a bitmap canvas consisting of 32×32 pixels. To draw a recognisable shape, line-drawing actions must be grouped together into shape-drawing actions that define shape boundaries. The agent is limited to drawing shapes when it sketches, i.e. it cannot draw arbitrary disconnected lines. This limits the possible sketches to combinations of closed rectilinear forms; see Figure 5.7 for an example. This simplifies the task of detecting emergent sub-shapes because they must be either bounded regions or combinations of bounded regions.

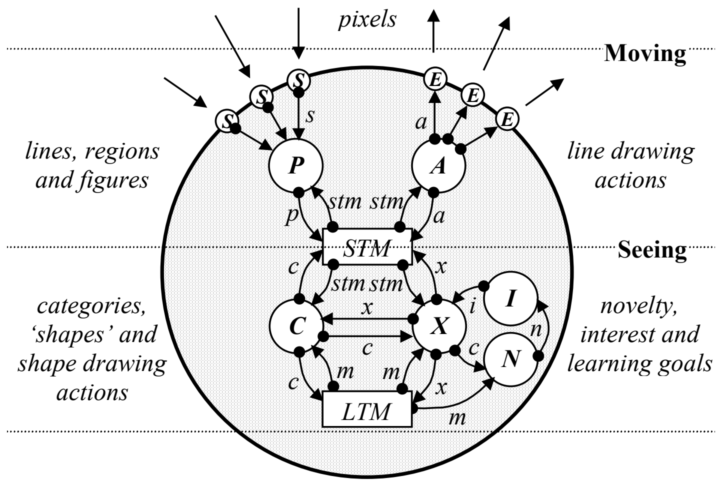


Figure 5.2: The architecture of the Reflect-a-Sketch agent divided into 'moving' and 'seeing' subsystems showing the representations constructed by each process.

5.2.2.3 Sensing

Sensed data are transformed by Reflect-a-Sketch's perceptual processes into a number of binary feature maps that represent the presence or absence of a feature at each pixel location. Reflect-a-Sketch perceives colours, lines and regions. Pixels of a specific colour are represented in a colour feature map. Reflect-a-Sketch can only draw using one colour, so the colour feature maps represent the foreground and background of the image. Horizontal and vertical line feature maps are produced using line detection algorithms based on (3×3) convolution matrices. Figure 5.3 illustrates the production of feature maps for foreground and background colours, horizontal and vertical lines for an example drawing. Above the horizontal and vertical line feature maps are the convolution matrices that produced them.

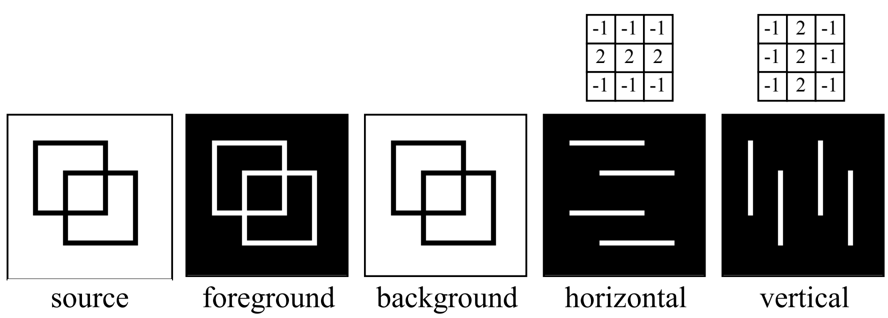


Figure 5.3: Example feature maps for foreground and background colours, vertical and horizontal lines (and the convolution matrices used to detect the lines).

5.2.2.4 Perceiving Regions

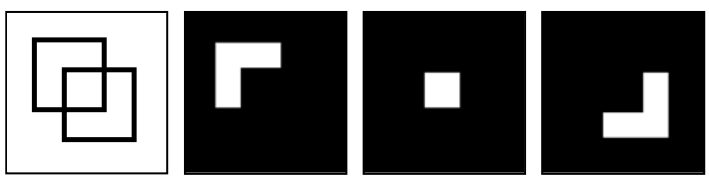
Reflect-a-Sketch uses a very simple yet effective algorithm to determine the bounded regions of a shape. The extraction of bounded regions is the basis of Reflect-a-

Sketch's ability to detect emergent shapes. The 'colouring book algorithm' is an extension of one of Ullman's visual routine to determine whether a point is on the inside or the outside of a figure (Ullman, 1984). The algorithm works by "colouring in" each region of a bitmap image by flood filling each one with a different colour. The colouring book algorithm is illustrated in Figure 5.4.

(a) (b) (c) (d) (e)

Figure 5.4: The colouring book algorithm used to find minimally bounded shapes in sketches. The paint can icons represent the locations of the flood fill commands used to fill white areas of the image as the algorithm scans from the top-left to the bottom-right corners. The top-left corner is assumed to be background and so the background is filled with black from this location to match with colour of the drawn lines (a & b). Each subsequent white pixel initiates a flood fill at that location with a different colour (c–e).

The algorithm begins by colouring in the background of the sketch by flood filling the image with the colour used to draw lines from the top-left corner. This removes the lines from potential inclusion as regions of interest. The colouring book algorithm continues by checking each pixel from the top-left corner to the bottom-right to determine whether it has already been filled. If a pixel has not been filled, a flood fill is started at that pixel location. A different fill colour is used for each flood fill operation, from the coloured image it is a simple matter to determine the minimally bounded regions by filtering the pixels by colour and separating each region into a feature map, as illustrated in Figure 5.5.



source region 1 region 2 region 3

Figure 5.5: The region feature maps extracted using the colouring book algorithm.

The colouring book algorithm extracts all of the regions that represent the minimally bounded sub-shapes in a drawing ready for the recognition of composite emergent shapes.

5.2.2.5 Perceiving Figures

Adding together the regions extracted by the colouring book algorithm and expanding the foreground area by a single pixel to compensate for the loss of the drawn lines produces a representation of a sketch's figure and ground. Figure 5.6 illustrates the process of constructing the figure for the regions extracted above.

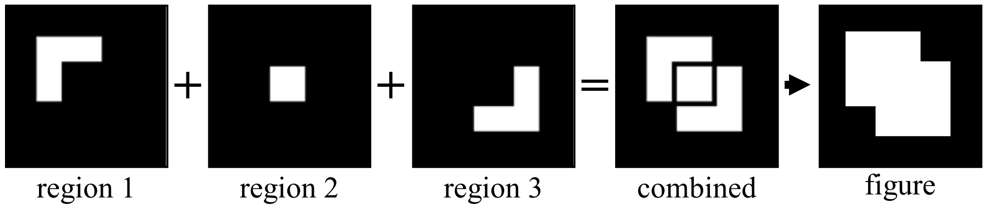


Figure 5.6: The construction of a representation of an images figure and ground from the bounded regions extracted by the colouring book algorithm.

5.2.2.6 Seeing

The regions extracted by the colouring book algorithm and the figures constructed from the combination of those regions provide the necessary representations to detect emergent sub-shapes and emergent super-shapes in the sketches produced by Reflect-a-Sketch. The 'seeing' subsystem of Reflect-a-Sketch processes these representations to produce categories for shapes and evaluate the novelty of the emergent shapes found.

5.2.2.7 Conceiving Shapes

Reflect-a-Sketch uses Self-Organising Maps (SOMs) to transform percepts into shapes. In this context, a 'shape' is a visual pattern that can be confidently assigned to a previously learned category. The input to each SOM is one of the feature maps computed by the perceptual processes converted into a vector of real numbers.

The output of each SOM is the activation of the best matching neuron in the map. The activation of the best matching neuron indicates the familiarity of the input pattern: the more familiar the pattern the greater the activation of the best matching neuron. The familiarity of a shape drawing action is estimated by simply tracking the number of times that the drawing action has been used. This does not require a learning system because no categorisation of the drawing action is required; shape-drawing actions are all uniquely identified within the memory of the agent.

5.2.2.8 Detecting Novelty

The familiarity of a drawing is used to determine its novelty. An unfamiliar drawing is novel by virtue of its atypicality. An unfamiliar drawing can also be surprising as long as the drawing actions that produced it are familiar. A familiar drawing can be surprising if the drawing actions that produced it are unfamiliar. An unfamiliar drawing produced by unfamiliar drawing actions is novel but not surprising. Reflect-a-Sketch can also determine another type of conflict when two categorisations do not agree, e.g. the categorisation of a section of the foreground does not agree with the categorisation of the enclosed bounded region. To measure the degree of conflict, each pixel in the reconstructed representation of the input is considered as an independent prediction of an expected feature. The degree of the conflict is calculated as the pixel-by-pixel difference between the constructed representations multiplied by the confidence of the categorisation.

5.2.2.9 Modelling Interest and Boredom

Interest is modelled using a linear hedonic function; the interest in a sketch is simply the novelty of that sketch. Reflect-a-Sketch also maintains an on-going measure of the interestingness of the design process. The agent's interest in the design process is calculated as the mean of its interest in the last 10 sketches. A minimum threshold on the agent's interest in the design process is used to model the onset of boredom. If the agent's interest falls below the boredom threshold the agent is determined to be under-stimulated or "bored" and seeks to find new stimulation through diversive exploration.

5.2.2.10 Learning Interesting Shapes

Reflect-a-Sketch learns to draw novel shapes that it finds interesting to expand its range of possible sketches. The process of extracting a bounded space region and constructing a new shape drawing action is illustrated in Figure 5.7. Reflect-a-Sketch can learn shape-drawing actions for novel regions and novel figures. When Reflect-a-Sketch discovers a novel figure it learns a shape-drawing action for the whole figure, including internal lines, so that it does not have to rediscover the interesting combination of primitive shapes to use it in the construction of more complex figures.

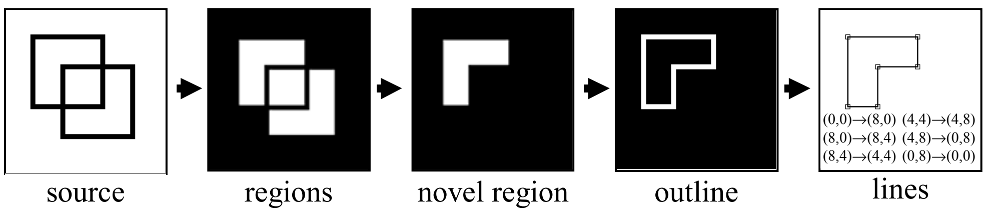


Figure 5.7: Construction of new shape drawing command from an extracted bounded space region.

Reflect-a-Sketch makes no attempt to generalise the shape-drawing actions other than to specify them relative to the top-left hand corner of the shape boundary to allow them to be drawn anywhere on the external canvas. This means that rotated versions of the same shape are considered unique, e.g. Region 1 and Region 3 in Figure 5.6 are learned as separate shape-drawing actions rather than rotated versions of the same shape. Consequently, Reflect-a-Sketch learns many more shapes than are strictly necessary, but can also explore the space of possible sketches with a simple drawing policy.

5.2.2.11 Exploring the Space of Sketches

Reflect-a-Sketch uses two very simple heuristics to control the production of new sketches based on its interest in recent sketches. These two heuristics model diversive and specific exploration respectively:

1) *Diversive exploration*: If interest falls below boredom threshold then increase the number of shapes drawn per sketch by 1.
2) *Specific exploration*: If the most recent sketch was so interesting as to have resulted in the learning of a new shape drawing action then reset the number of shapes per sketch to 1 and use the new shape drawing action.

The first heuristic increases the number of shapes used per sketch in an attempt to promote stimulation and produce new emergent shapes as a consequence of the interactions between shapes drawn in the same sketch. The second heuristic restarts the search for new emergent shapes using one instance of the most recently learned, i.e. interesting, shape as a starting point. This reduces the potential stimulation by reducing the number of shapes drawn while also allowing the agent to focus on the specifics of the new shape to learn perceptual categories.

5.2.3 Results

A screenshot of an early implementation of Reflect-a-Sketch is shown in Figure 5.8. The figure shows the extraction of a region of interest into the sensory and perceptual processes labelled "Literal Apprehension" and "Emergent (Space) Apprehension" respectively, after Schön and Wiggins (1992). This version of the program can only detect emergent sub-shapes; it cannot detect emergent figures and so is not able to construct complex sketches consisting of multiple instances of composite shape-drawing actions. Despite this it was still able to produce some complex sketches having started with a single shape-drawing action for an 8x8 pixel square.

Figure 5.8 shows that the inputs and the representations constructed by the literal and bounded-space learning systems are in conflict: the input bounded space representation shows a rotated L-shape while the learning systems have both constructed equivalent representations of a square. This is a significant mismatch and indicates the degree of novelty of the situation. Novelty is measured in these circumstances as a value proportional to the highest confidence of the two learning systems, which in this case is the confidence of the literal apprehension learning system.

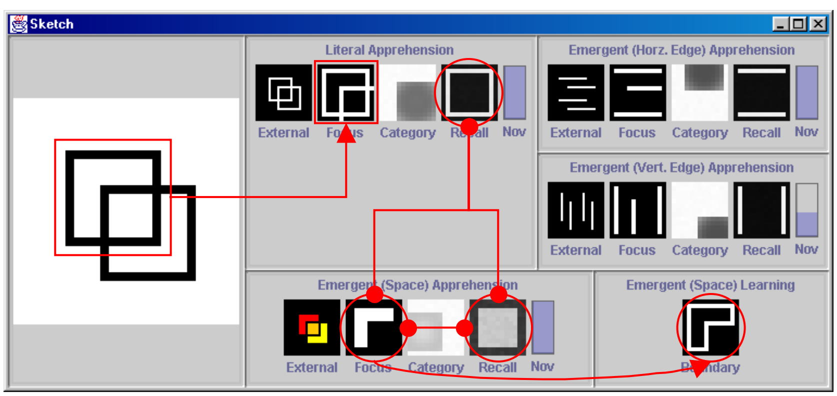


Figure 5.8: A screenshot of an early implementation of Reflect-a-Sketch in operation showing the current design and the different representations constructed.

Figure 5.8 also shows that the emergent L-shaped region perceived by the bounded space processes has been analysed and an image of the boundary of the emergent shape has been generated. Figure 5.9 illustrates a typical progression of emergent shapes learned by Reflect-a-Sketch over the course of a short run.

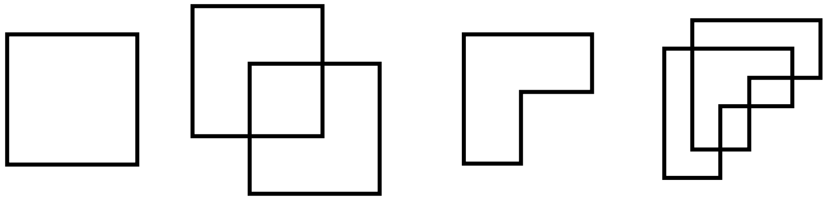


Figure 5.9: A typical progression of drawings considered interesting by Reflect-a-Sketch.

Figure 5.10 is a screenshot of the final version of Reflect-a-Sketch. The major difference between the two versions is that the final version includes representations of the figure as well as the bounded sub-shapes. The displays for each representation include a trace of recently detected novelty.

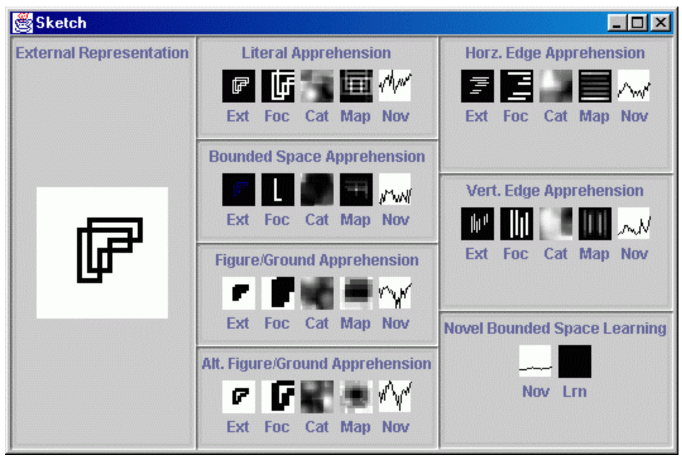


Figure 5.10: A screenshot of the final version of Reflect-a-Sketch in operation showing all of the representations built for a complex sketch.

Two "styles" of sketches emerge from the explorations of Reflect-a-Sketch; firstly, there are sketches with complex internal structures favoured by an interest in emergent sub-shapes; and secondly, there sketches with complex boundaries favoured by an interest in emergent super-shapes.

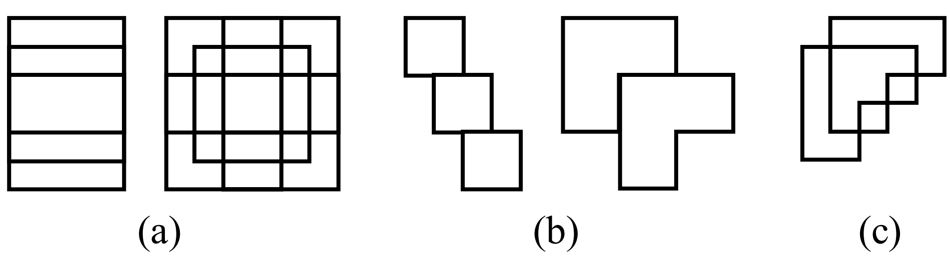


Figure 5.11: Some "interesting" sketches produced by Reflect-a-Sketch: (a) emergent subshapes, (b) emergent super-shapes, and (c) emergent sub-shapes and super-shapes.

Figure 5.12 illustrates the curious learning behaviour of Reflect-a-Sketch as the number of shape-drawing actions learned over time. The steps in Figure 5.12 show that the agent learns new drawing actions in short bursts when an interesting sketch is produced, indicating that several emergent shapes tend to appear together. The initially long plateau indicates the early learning process when the SOMs are establishing an initial arrangement and the confidence in all predictions is low, resulting in little or no novelty being detected.

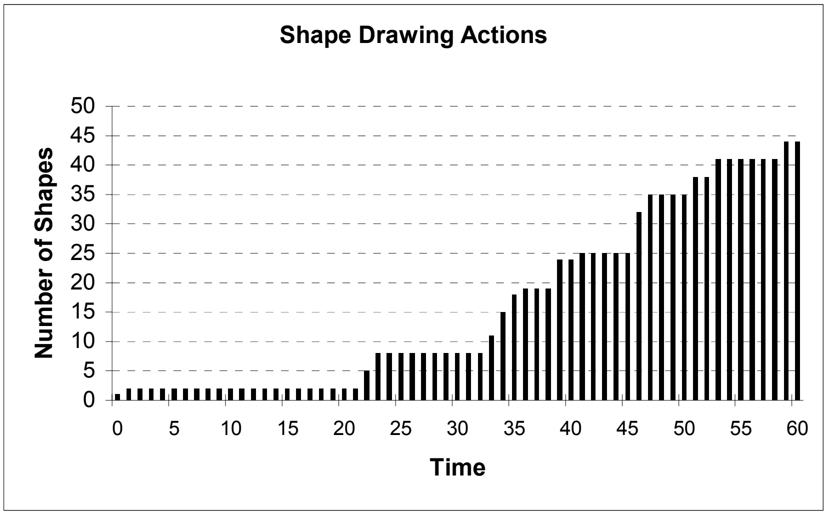


Figure 5.12: The number of shape drawing actions learned by Reflect-a-Sketch over the first 60 time steps of a run.

5.2.4 Analysis

The search for novelty promotes the construction of complex forms in Reflect-a-Sketch. The nature of these forms depends upon the type of novel shapes that the agent is searching for. An interest in emergent subshapes leads to a development of forms with complex internal structures, often with simple boundaries. An emphasis on novel emergent figures leads to complex boundaries, sometimes constructed using simple components.

In both cases, the curious agent quickly produces forms that are beyond its limited ability to comprehend and the agent does not find any more interesting forms. In the first case, the internal structures often become so complex that the only subshapes to be found are small rectangles of which the agent quickly learns a full catalogue. In the second case, the agent quickly produces figures that extend beyond the bounds of the region of interest and so cannot be comprehended as bounded shapes to be learned by the agent.

5.2.5 Discussion

Reflect-a-Sketch extends previous computational work in shape emergence by adding a learning component that can come to expect emergent shapes. This is an important aspect of the curious behaviour of the agent, guiding the process of learning new drawing skills and hence the expansion of design spaces. Importantly, Reflect-a-Sketch quickly expands its repertoire of drawing skills and produces drawings that were impossible given its initial skill set. The inclusion of a curious component makes the process of emergence more like that described by Gero as a creative design process (Gero, 1994b).

5.2.6 Conclusion

Reflect-a-Sketch successfully demonstrates the role that curiosity can play in the direct manipulation of material. However, the processes implementing curiosity are few (and simple) compared to the processes that provide it with the data. The level of sophistication required in the systems implementing the 'moving' aspects of the reflective sketching process makes it an awkward vehicle for the study curious behaviour. Possible future research directions using a similar architecture to Reflect-a-Sketch are discussed in Chapter 7.

To continue research into the curiosity it was decided that, like designers in the real world, curious agents should use more abstract descriptions of the design artefact in their explorations of design space. Hence, the following agent uses a parametric design method to explore the space of Spirograph patterns.

5.3 A SPIROGRAPH EXPLORER

This section presents some experiments with an agent that explores the parametric design space of patterns generated using a simulated Spirograph. The goals of this experiment are to examine the behaviour of a curious agent as it explores a parametric design space and to illustrate the difference in the representations built by curious and non-curious design agents.

5.3.1 Motivation

The Spirograph[2] epitomises a simple parametric design tool. Spirograph sets consist of an array of plastic gears. To draw a pattern one gear is fixed to a piece of paper and a second gear is moved around it while tracing its path by pushing a pen through a hole in the interior. This wonderfully simple toy has charmed children for over 30 years and has no doubt sparked an interest for geometrical patterns in budding architects and designers.

5.3.1.1 The Geometric Lathe

Long before the Spirograph was created to entertain, similar technology was being used to generate complex geometrical patterns for the far more serious purpose of defeating forgery of banknotes.

Complex geometrical patterns have been used as anti-forgery devices on banknotes since the 19th Century. The machine used to generate the complex patterns was called the Geometric Lathe; its invention was heralded as a major breakthrough in financial security:

> [...] the Geometric Lathe has been esteemed, at all times, as the sheet anchor of public security against the dangers of forgery. […] The least change of a wheel of the eccentric, or turn of a set screw, produces a new pattern that shames the kaleidoscope. It defies the efforts of the mathematician to calculate the extent of its variations; […] and when the transfer press is brought to its aid, […] human ingenuity fails in the attempt to produce an imitation.

[2] Spirograph is a registered trademark of Hasbro.

Extracted from: "Remarks on the Manufacture of Bank Notes and other Promises to Pay: Addressed to the Bankers of the Southern Confederacy" (1864) [3]

Figure 5.13 displays a $5,000 U.S. tax proof approved in 1872 that is a wonderful example of the intricate designs that were produced using the Geometric Lathe.

Figure 5.13: A proof for a $5,000 U.S. tax stamp approved in 1872 for use in the taxation of a multi-million dollar expansion of the Union Pacific Railroads but never used. It is considered one of the finest examples of its kind.

The security provided by the patterns produced by the Geometric Lathe lay in the difficulty of reversing the process used to generate them, i.e. determining the correct settings for the Geometric Lathe to produce a desired pattern. Modern encryption technologies work in similar way by exploiting the difficulty of the process of factoring large numbers work (Rivest et al., 1978).

Although more modern security devices have superseded the geometric patterns as the primary means of securing important documents against forgery, these patterns can still be found on many banknotes, as well as other important documents such as stocks, bonds, and passports.

The use of the Geometric Lathe is a classic example of using emergence in design. The security offered by the patterns came primarily from the emergence of distinctive patterns from the intricate latticework of lines. Perhaps more than any other example of design, the design of these patterns relied on the ability of the designer to determine the unexpectedness of the patterns generated. The agent presented in this section attempts to mimic this search in the simpler domain of Spirograph patterns.

[3] Available on the Internet at http://docsouth.unc.edu/banknote/frontis.html

5.3.2 Implementation[4]

The pattern that a Spirograph generates when one wheel is rotated around another is known in mathematics as an *epicycloid*. The mathematics of epicycloids is illustrated in Figure 5.14.

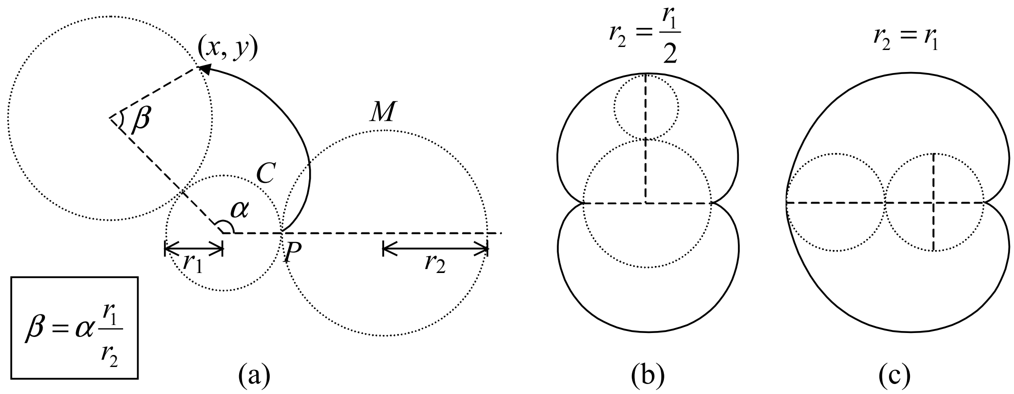


Figure 5.14: The mathematics of epicycloids: (a) path of epicycloid shown along arc from its starting point *P* to (*x*, *y*) at α around fixed gear *C* of radius r_1 and β around moving gear *M* of radius r_2, (b) epicycloid generated with $r_2 = r_1 / 2$, and (c) epicycloid generated with $r_2 = r_1$.

Spirograph patterns generated using an arrangement of circular gears can be mathematically modelled as epicycloids using the following equations:

$$
\begin{aligned}
&x = (r_1 + r_2) \times \cos\alpha - r_2 \times \cos(\alpha + \beta) \\
&y = (r_1 + r_2) \times \sin\alpha - r_2 \times \sin(\alpha + \beta) \\
&\text{where:} \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad (5.23) \\
&r_1 = \text{radius of fixed gear} \qquad \alpha = \text{rotation around fixed gear} \\
&r_2 = \text{radius of moving gear} \qquad \beta = \text{rotation around moving gear}
\end{aligned}
$$

5.3.2.1 Typical Spirograph Patterns

Figure 5.15 illustrates the variety of Spirograph patterns that is possible for a small selection of random values for r_1 and r_2.

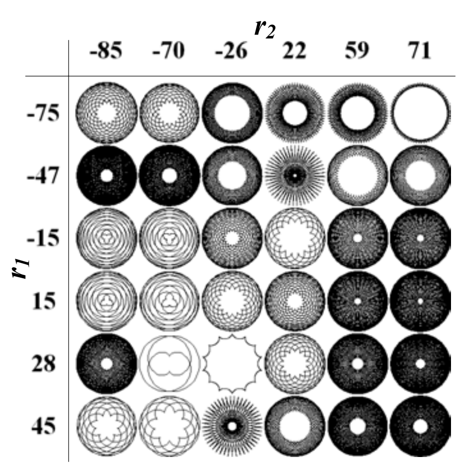


Figure 5.15: A random sample of Spirograph patterns with a small selection of random values for the fixed gear radius (r_1) and the moving gear radius (r_2).

[4] The Spirograph pattern generator described in this section can be found on the accompanying CD-ROM: see Appendix B for details.

Visually, two broad categories of Spirograph patterns can be distinguished in Figure 5.15: simple patterns produced with a few rotations of the moving gear around the fixed gear and complex patterns produced with many rotations. The number of rotations is dictated by the greatest common denominator of the two radii. Figure 5.15 shows that for a random sample of the design space, complex patterns are far more common.

5.3.2.2 A Curious Spirograph Explorer

The architecture of the curious agent used in this experiment is illustrated in Figure 5.16. The high level functions of conception and curiosity are similar to those in Reflect-a-Sketch. The processes for perception and action are far simpler than those for Reflect-a-Sketch. Perception simply converts sensed data into inputs suitable for categorisation. Action issues parameter change commands to the effectors.

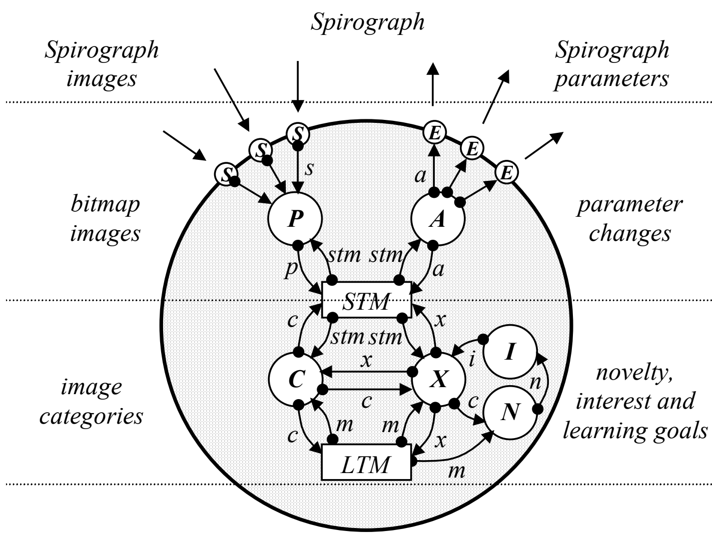


Figure 5.16: The curious agent for Spirograph exploration.

5.3.2.3 Parametric Design Actions

A curious design agent was set the task of searching a sub-space of possible Spirograph patterns bounded by gear ratios r_1:r_2 from -100:1 to 100:1. The agent explored the space of possible Spirograph patterns by changing the value of the ratio directly, rather than the values of the gear radii, as this provided a more predictable space of patterns because similar gear radii can produce wildly different patterns, whereas similar ratios generally produce similar patterns. The patterns were analysed by the agent as 32x32 pixel greyscale images. At this resolution many of the finer details of the patterns are lost and the classification of the patterns is based on larger scale features.

5.3.2.4 Spirograph Pattern Exploration

The curious design agent explores the space of Spirograph patterns by varying the amount that the gear ratio is changed at each step. When the agent finds interesting, i.e. novel, patterns it reduces the gear ratio to explore the space more thoroughly. When the agent finds patterns that are uninteresting it increases the gear ratio to quickly explore other design subspaces. The novelty of a pattern is detected using a

SOM as described in the previous chapter. The agent uses a linear hedonic function to determine interestingness from detected novelty.

5.3.3 Results

The novelty detected by the curious agent during a search of the design space for the first 200 time steps is shown in Figure 5.17. The chart shows that the curious design agent detects little significant novelty while performing an initial search of the design space, i.e. up to time step 55 marked on the chart. The lack of sustained interest in any patterns is not surprising; the agent has no categories of typical patterns with which to compare new patterns to determine novelty. After it has constructed some typical pattern categories it can start to be interested in novel patterns. The agent then repeatedly finds interesting patterns, indicated by peaks in novelty and will remain with a novel pattern to learn a category for it, producing a tailing-off in the novelty detected as the category is constructed.

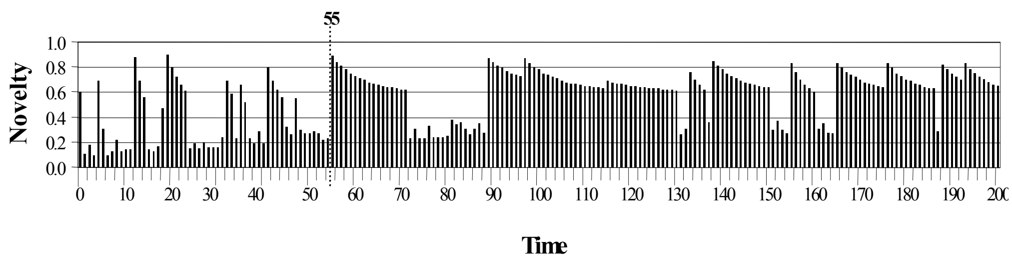


Figure 5.17: A chart showing the novelty detected over time for the first 200 time steps of a typical run. The chart shows an initial period where little novelty is detected followed by a series of peaks as novel patterns are discovered. The chart also shows the tailing-off of the novelty detected after each peak as the novel pattern's category is learned.

5.3.4 Analysis

To better understand the behaviour of this curious design agent it is useful to take a look at the representations of the design space that it constructs during exploration. This section compares the representation of a curious and a non-curious agent exploring the space of Spirograph patterns.

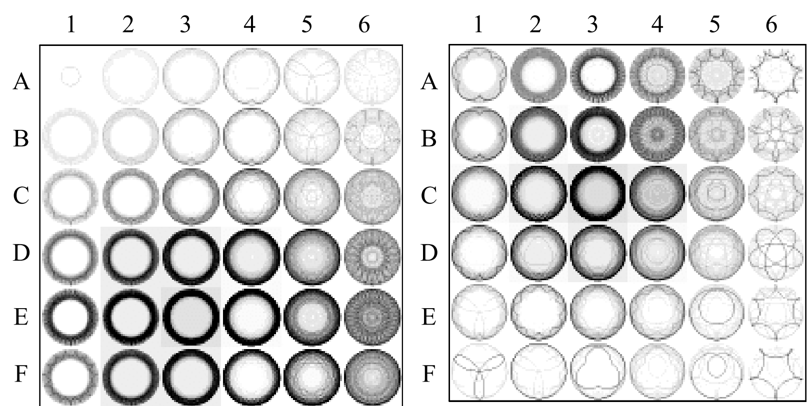
5.3.4.1 Visualising Curious Representations

SOMs are often used to visualise complex multi-dimensional vector spaces. To visualise the differences between representations built using curious and non-curious agents two agents were set the task of learning the space of Spirograph patterns. The non-curious agent explored the space of patterns by performing a "random walk" through the possible ratio values, perturbing the ratio slightly at every step. The curious agent explored the space as described above. Each agent explored the design space for 400 time steps.

Figure 5.18 shows the prototype representations learned by the SOMs of the two agents. Each grid shows the two-dimensional map of the design space that has been learned. Each cell corresponds to a neuron in the lattice of the SOM and shows the prototypical image of the category it represents. The prototype is an average of the images of the patterns that are contained within the category.

Although the networks have mapped the design space differently, some correspondences can be found, e.g. the categories found in the bottom half of the non-curious agent's map (D1–F6) roughly correspond to those found in the top-left corner of the curious agent's map (A1–D4).

The maps share similar representations for typical patterns, e.g. compare the prototypes E1, E5, F5 in the non-curious agent's map with A3, C4, D3 in the curious agent's map respectively. However, the maps differ considerably in their representation of novel patterns. For example, the non-curious agent's map has only a few categories in rows A and B that are similar to those found in rows E and F and columns 5 and 6 of the curious agent's map. In fact, the curious agent's map devotes nearly 50% of its categories to different types of novel patterns whereas only ~15% of the categories in the non-curious agent's map represent unique novel patterns.



(a) Random selection (b) Curious selection

Figure 5.18: The above maps show the bitmap images of the prototype patterns represented by the neurons of two 6x6 SOMs trained with (a) a random selection of patterns chosen by a non-curious agent, and (b) a set of 'interesting' patterns chosen by a curious design agent.

5.3.5 Discussion

Figure 5.17 shows that little novelty is detected in the first 50 time steps of searching. During this phase the lack of novelty means that mutations of the design parameters, i.e. the ratio of gears, will be high. Consequently, the agent begins by learning from a fairly random sample of patterns and as Figure 5.15 illustrates the agent is likely to be exposed to far more complex patterns than simple ones. The result is that the agent learns a set of categories for typical, complex patterns first.

Once an initial set of pattern categories has been learned, novel patterns, i.e. simple patterns, can be recognized. Figure 5.17 shows that the agent spends most of its time beyond the time step 55 learning novel patterns in some cases spending as long as 10 time steps learning a single pattern, indicated by the slow decay in the novelty detected. The two phases of learning are reflected in the differences between the SOMs produced by the curious and non-curious agents shown in Figure 5.18. The differences in the representation of atypical patterns between the maps reflect the prolonged learning given to novel patterns as a consequence the actions of the curious agent.

The improved representation of novel patterns in the curious agent's map means that it will no longer find these patterns to be novel and will have to search for new patterns in order to maintain interest in the design space. The neural networks used in the above experiment are relatively small compared to the design space, and so are never likely to accurately represent all of the possible Spirograph patterns. However, larger neural networks or agents that use ART networks may be able to learn representations for the entire space of Spirograph patterns so well that they can become "bored" with the entire space, potentially triggering the exploration of new design spaces.

5.3.6 Conclusions

Unlike Reflect-a-Sketch, the curious Spirograph explorer does not have the capability of expanding the design space it is given; instead it identifies the most interesting subspaces to concentrate its learning efforts upon. Consequently, the agent presented here models possible innovation within the space, finding unexpected patterns, but not creativity as defined by Gero (1994b). To model more creative forms of designing the agent would have to be given the opportunity to expand the ranges and number of design variables. This takes the agent beyond the scope of simple parametric design towards design methods such as Dimensional Variable Expansion, proposed by Cagan and Agogino (1991).

Many design problems can be cast as parametric design problems and curious design agents such as the one presented here may be useful in exploring the parametric design spaces for interesting, atypical designs; particularly when the number of parameters is large enough to make manual exploration tedious. Another example of a curious design agent exploring parametric design spaces is given in the next chapter. The following application presents a different way to decompose the design task between agent and tool.

5.4 CURIOUS EVOLUTION

The final study in this chapter demonstrates the ability of a curious design agent to guide the design process of an evolutionary design system. Unlike the previous two applications, the curious design agent presented here does not manipulate variables that directly affect products. Instead it issues commands, in the form of selection preferences, to an interactive evolutionary design system to affect the design process. The curious design agent provides judgements of interestingness in much the same way that human users do when using interactive evolutionary design tools (Sims, 1991; Todd and Latham, 1992).

5.4.1 Background

In "The Clockwork Muse" Martindale (1990) presented an extensive investigation into the role that the search for novelty plays in literature, music, visual arts and architecture. He concluded that the search for novelty exerts a powerful force on the development of artistic movements. Martindale illustrated the influence that the search for novelty has on creative activity with the following thought experiment, "The Law of Novelty":

> We live in a predictable world. […] Every morning we are bombarded with automatic "Good mornings" and routine inquiries about how we are. Our days are full of pat questions and equally pat replies. Imagine what would happen, however, if some of us, tiring of this state of affairs, decide to do something about it. Decide, indeed, to outlaw any and all repetition. Once something has been said, it can never be said again. Once something has been done, it cannot be done again. This requires no act of Congress. We can implement it ourselves. We can, as well, impose a sanction more severe than the death penalty. Anyone who says something that has already been said, or does something that has already been done, will simply be ignored. Someone who persists in the crime of repetition will find that he or she ceases to exist. No one will pay the slightest attention to the person.

Martindale elaborated some of the consequences of adopting his Law of Novelty. For example, discourse would be forced into ever more concrete and specific forms to ensure that the meaning is understood while at the same time driving speakers to circumlocution in order to avoid repetition. For example, a salesman's description of a toaster becomes an exercise in metaphors: "Rather than saying, 'This is our newest model,' something such as 'This is the rising sun of the destroyers of leavened moisture,' is called for."

Martindale argued that what he described was not a revolution but merely a magnification of the world we live in and that in fact the Law of Novelty is applied in its purest form in the arts. Exact replication is not allowed in any of the arts: otherwise the notion of forgery would be nonsense. And although artists may not be ignored while they are alive the rule of novelty takes effect with exceptionless brutality upon their deaths. It is then that they begin to be ignored.

As well as demonstrating the application of curious design agents in the control of a design tool, this section presents an attempt to implement the Law of Novelty, for a single design agent – investigations of social applications of the Law of Novelty are deferred until Chapter 6 within the context of an artificial creativity simulation. The aim of the work presented here is to explore the relationship between searching for novelty, i.e. curious exploration, and elaboration of design products into increasing complex forms to satisfy the need for continual innovation.

5.4.2 Implementation[5]

Interactive evolutionary systems have become a popular tool for exploring aesthetic design spaces because they allow a human operator to generate complex designs without the need to manipulate design variables. The pioneers of interactive evolutionary systems include Richard Dawkins, Karl Sims, and William Latham (Dawkins, 1987; Sims, 1991; Todd and Latham, 1992).

Dawkins first popularised this method of evolving aesthetically pleasing images in his book "The Blind Watchmaker" (Dawkins, 1987) with a program that evolved "biomorphs" – small stick figures that resembled insects, butterflies or trees

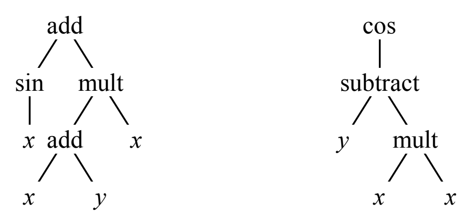
[5] An implementation of the interactive evolutionary system used by the curious design agents is provided on the CD-ROM: see Appendix B for details.

depending on the specifics of the evolved genes[6]. Todd and Latham (1992) developed a much more complex form of interactive evolution to allow William Latham to evolve "virtual sculptures" that he has exhibited in galleries worldwide. Karl Sims developed an interactive evolutionary system that was capable of producing stunning artworks using techniques borrowed from genetic programming (Sims, 1991).

Interactive evolutionary art systems are standard evolutionary systems that rely on user interaction to provide subjective evaluations about the 'fitness' of evolved artworks. The users of an evolutionary art system evaluate artworks based on aesthetic preference to guide the evolutionary process towards the generation of more aesthetically pleasing artworks. In general, interactive evolutionary systems have very small population sizes compared to standard evolutionary systems; only 9–16 individuals per population, compared to 100s or even 1000s per population in a standard evolutionary system.

5.4.2.1 The Evolution of Images

Karl Sims is probably best known for his work developing one of the first interactive evolutionary art systems for complex two-dimensional bitmap images (Sims, 1991). Using Genetic Programming (Koza, 1992), Sims devised an evolutionary art system that produced artworks by evolving symbolic function trees. Two simple function trees are illustrated in Figure 5.19.



(add (sin *x*) (mult (add *x y*) *x*)) (cos (subtract (mult *x x*) *y*))

Figure 5.19: Two example function trees and their corresponding Lisp expressions that can be used to generate genetic artworks.

Two genetic operators called "crossover" and "mutation" are used to generate new individuals from a pair of selected 'parents'. The crossover operator exchanges a randomly chosen branch of one parent tree with a randomly chosen branch of the other parent to generate a child. Figure 5.20 shows four possible children of the two equations shown in Figure 5.19.

The mutation operator requires only a single parent. It removes a branch of the parent and replaces it with a randomly 'grown' branch. The growth process begins with a spare node produced by the removal of an existing branch. It replaces the spare node with a randomly chosen operator (e.g. add, mult, sin, etc.) or terminal (e.g. *x*, *y*, etc.). When an operator is chosen a number of new spare nodes are created, one for each required input of the operator. The process repeatedly replaces spare nodes until they are all filled. Figure 5.21 shows four possible children produced by mutating the first equation shown in Figure 5.19.

[6] See Dawkins (1989) for an interesting discussion on the evolution of evolvability detailing the development of the biomorph program.

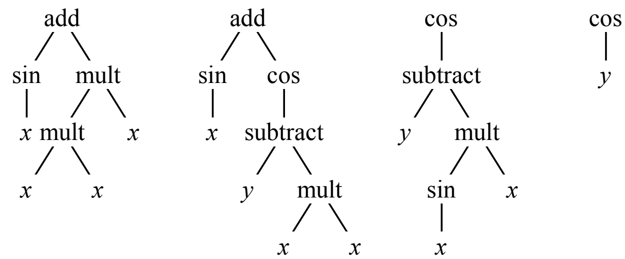


Figure 5.20: Four possible children of (add (sin *x*) (mult (add *x y*) *x*)) and (cos (subtract (mult *x x*) *y*)). In the first two children the first equation received a branch from the second equation and in the second two children the first equation donated a branch.

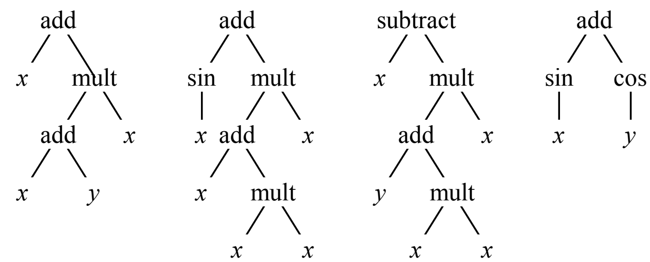


Figure 5.21: Four possible children produced by mutating the parent (add (sin *x*) (mult (add *x y*) *x*)). Mutating the root node of the parent produced the third example where the entire tree has been replaced.

5.4.2.2 Genetic Artworks

To produce an image a symbolic function tree is evaluated at a set of points, typically between (-0.5, -0.5) and (0.5, 0.5), that corresponds to every pixel location in an image of a given size. The values for *x* and *y* at the terminal nodes of the tree are substituted with the x and y co-ordinates of the sample point and the values for every higher node in the tree are recalculated. The root node evaluation is then interpreted to produce a colour value that is assigned to the pixels of the output image.

An example genetic artwork is shown in Figure 5.22; human participants evolved this particular genetic artwork over the Internet as part of the International Interactive Genetic Art (IIGA) project (Witbrock and Reilly, 1999). The code used to evolve images in the IIGA project is used in this application to evolve genetic artworks.

Unlike the work of Sims, which uses a rich mix of computer graphics procedures and image processing techniques, the genetic art systems of the IIGA project use quaternion mathematics that deal with four-dimensional numbers. The result of evaluating a quaternion expression is a four-dimensional number that can be transformed into a three-dimensional colour vector.

Figure 5.22: An example of a genetic artwork interactively evolved by a human user. (From the archive of evolved genetic artworks in Interactive Genetic Art III.)

5.4.2.3 A Curious Evolver

A curious design agent, or "curious evolver", controls the interactive evolutionary algorithm; it guides the evolutionary process by selecting interesting images in much the same way that human users guide interactive evolutionary algorithms. The images selected by the curious evolver are called artworks throughout this section and in the following chapter — the images are considered to be artworks because they play the role of artworks in the simulation. In this example they play the role of interesting images according to the subjective view of the agent and thus stimulate further exploration. In the system described in Section 6.2 interesting images play a role in sole creativity by acting as a form of currency between communicating agents. Importantly, the images are not considered artworks because they have any claim to artistic merit by human standards. The architecture for a curious evolver is illustrated in Figure 5.23.

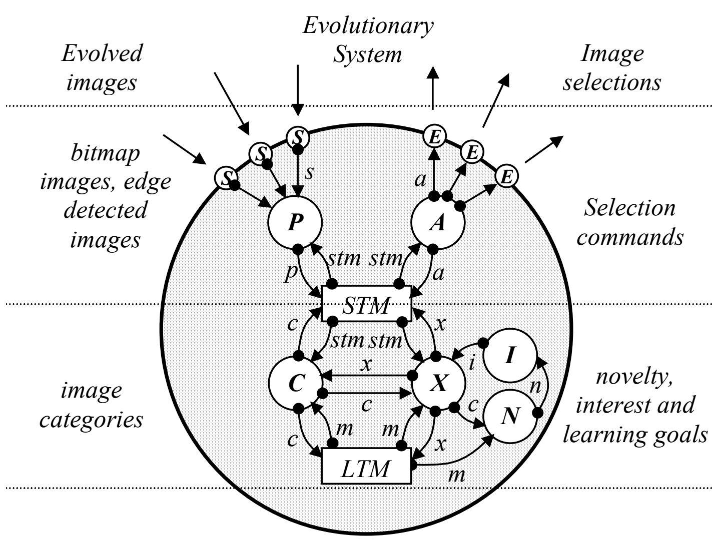


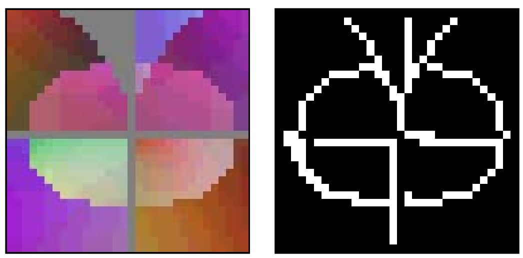
Figure 5.23: Architecture of a curious evolver.

The architectures of a curious evolver and the curious Spirograph explorer, illustrated in Figure 5.16, are very similar, especially at the higher levels of conception and curiosity. The most significant differences between the two agent architectures are the differences in the types of actions taken and some additional image processing in the perception of evolved images. All of the other differences between the two applications are to be found in the external environment of the curious evolver, i.e. the interactive evolutionary system described above.

5.4.2.4 Image Processing

A 32x32-pixel image of each genetic artwork is produced for analysis and categorisation in order to determine its novelty. Although this is a low-resolution image it is still large enough to allow fairly complex artworks to be evolved.

A relatively simple combination of a Laplacian edge-detector and a fixed intensity threshold function were used to extract a binary image of the predominant edges in an artwork, as shown in Figure 5.24.



(a) (b)

Figure 5.24: The image processing applied to genetic artworks to extract the edge structure of the images, (a) the original image, and (b) the binary image produced by the image processing to find the most prominent edges.

5.4.2.5 Novelty Detection

Long-term memory is implemented as a SOM. A SOM containing a lattice of 6x6 map neurons is used to provide a memory of previous image as prototypes. The input to the SOM is the 32x32 pixel image. The relatively small size of the SOM means that the network provides a medium-term rather than long-term memory. Prototypes are "forgotten" as new areas of the design space are explored – see Section 4.2.5.2 for an example of a SOM being dragged across an input space.

As the design space is explored, the SOM produces neighbourhoods of similar prototypes to capture the variations in a particular image type. Figure 5.25 shows the neighbourhoods that have formed for similar input patterns, e.g. around E2 and A5. The bridging of unseen areas of the design space, as demonstrated in Section 4.2.5.1 can be seen in the mixing of the neighbourhood patterns in the intermediate areas, e.g. around D4.

As in the previous applications of SOM-based curious agents, the novelty of each new image is measured as the distance between it and the nearest matching prototype. The distance is defined as the Euclidean distance between the vectors representing the new image and the closest matching prototype in the 1024 dimensional input space. The novelty values reported in the remainder of this section are the raw novelty values, i.e. the values of output by the best matching neuron of the neural network. For the size of image used these values range between N=0 and N=32, with N=0 being an exact match and N=32 being a complete mismatch. Interest is calculated using a Wundt curve hedonic function as described in Section 2.4.2.

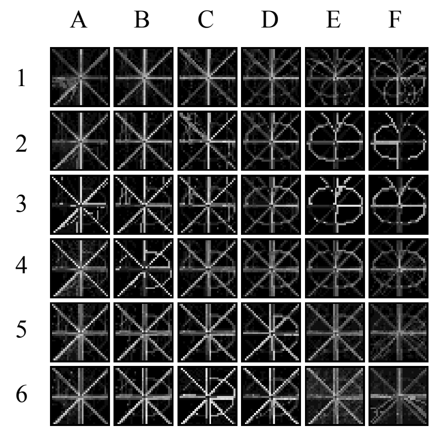


Figure 5.25: The prototypes represented by the 36 neurons of a self-organising map having just categorised the input shown in Figure 4b at location E2.

5.4.2.6 Exploration

Curious exploration is implemented by allowing the agent to select the most interesting artwork in each generation to be the parent of the next generation. Each generation consists of the parent artwork and eight new artworks generated using only the mutation operator — evolution is restricted to using mutation only because this simplifies the interaction between an agent and an evolutionary system. The curious evolver evaluates the novelty of each member of the population before it begins to learn any new representations for them to ensure that the order in which the artworks within a population are evaluated does not affect the novelty measure assigned to them. This means that the curious evolver differs slightly from the previous design agents in that it is able to turn on and off its learning functions to enable a fair comparison between artworks presented simultaneously.

5.4.3 Results

To investigate the relationship between the search for novelty and the complexity of resulting artworks an experiment was conducted to compare agents with different preferences for novelty encoded in their hedonic functions. To measure the complexity of the images the fractal dimension of selected images was calculated. The calculation was performed on the images after image processing to determine the dominant edges so that the fractal dimension would be that of the images as perceived by the agents. The fractal dimension was estimated using the box counting method – this is the same method that Taylor et al. (1999) used to determine the fractal dimension of Jackson Pollock's drip paintings.

For any two-dimensional image, a measure of its fractal dimension will produce a value between 0.0 and 2.0, depending on how much of the space is filled in the image at different levels of detail. To calculate the fractal dimension of an image a series of grids are placed over the image and the number of boxes occupied by the feature of interest in the image is counted. Figure 5.26 illustrates the process for the image-processed artwork shown in Figure 5.24 where the edge segments are the feature of interest.

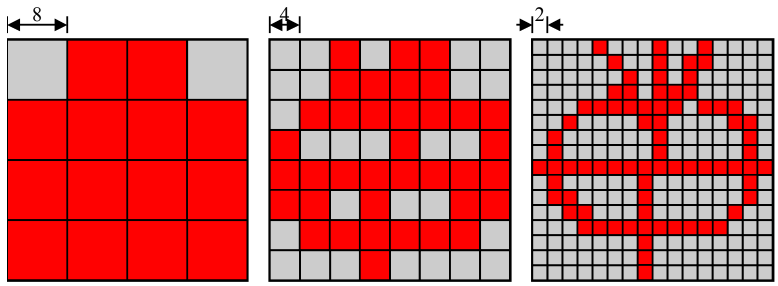


Figure 5.26: The box-counting method of estimating the fractal dimension of an image.

The fractal dimension can be calculated manually by plotting the count of boxes containing features against the number of boxes per side on a log-log graph and performing a linear regression. The gradient of the line produced is used as an estimate of the fractal dimension. More information about the box-counting method of fractal dimension estimation can be found in Mandelbrot (1977).

To investigate the relationship between the preferred degree of novelty and the fractal dimension of the resulting images, two types of agents were used. One type preferred novelty values of N=18 and the other type favoured novelty values of N=11. Three agents of each type were allowed to explore the space of genetic artworks for 50 time steps.

Figure 5.27 shows how the average fractal dimension of the images selected by the three agents in each test group changed over time. The graph shows that agents with a preference for greater novelty produce images with higher fractal dimensions, appearing to confirm Martindale's hypothesis.

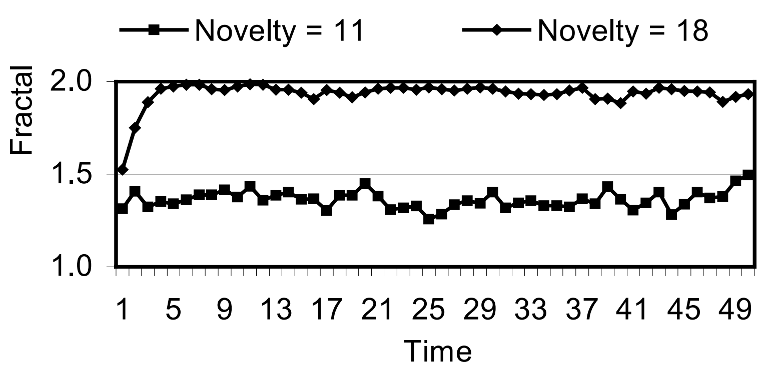


Figure 5.27: The development of two distinct styles of images with different fractal dimensions in two groups of agents with hedonic functions that peak for the values of novelty indicated.

To confirm this relationship between fractal dimension and preferred novelty, similar tests (3 agents/group for 50 time steps) were performed for a total of 19 different test groups with hedonic functions that favoured novelty values in the range $1 \leq N \leq 19$. Figure 5.28 shows that the relationship between the preferred value of novelty and the fractal dimension of the resulting images is almost linear for the large proportion of values for preferred novelty. Performing a linear regression on the data points we discover that on average the fractal dimension of the resulting image goes up by 0.1 per unit step in novelty preferred.

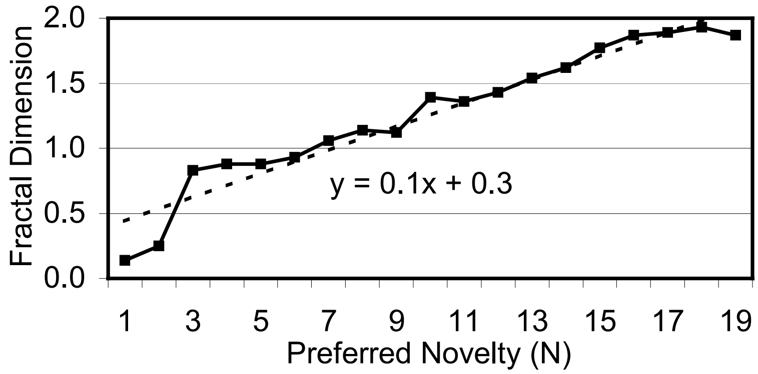


Figure 5.28: A comparison of the average values for 3 agents of the fractal dimension of evolved images after 50 time steps against a range of peak hedonic values. The equation and dashed line show the result of performing a linear regression on the sample points.

Visually this means that the images produced by agents that prefer greater novelty appear more complicated than those produced by agents that prefer lower amounts of novelty. Figure 5.29 displays a small gallery of images recorded as examples of interesting artworks by the test groups with preference for the novelty.

5.4.4 Analysis

How can we explain this relationship between the preferred novelty of an agent and the fractal dimension of the resulting images? One explanation is that the curious exploration of the space of genetic artworks drives the agents towards subspaces that have an appropriate amount of local variability to continually satisfy the need for novelty. Consequently, agents that prefer novel forms will tend towards areas of the design space that produce more complex images, as there is a great deal more variability between complex images than between simple ones.

5.4.5 Discussion

One of the reasons why interactive evolutionary design systems have become popular is that they allow a person to explore a design domain without having to understand the mechanisms of either the design generation or the evolutionary process. The simple architecture for the design agents used in these experiments, shows that curious design agents can take advantage of the same decoupling of generation and test in the design cycle and concentrate on producing interesting designs without the complications of having to control the details of how the designs are produced.

A potential advantage of using a design tool, such as an interactive evolutionary system, is that it should allow a single agent to explore multiple domains without learning new design skills. The design process implemented by the design tool allows an agent to explore multiple domains using the same set of instructions. In other words, the design agent does not need to understand the details of the domain variables in order to explore design spaces. The potential for analogical forms of design reasoning emerging from such inter-domain exploration make this an interesting direction for future research.

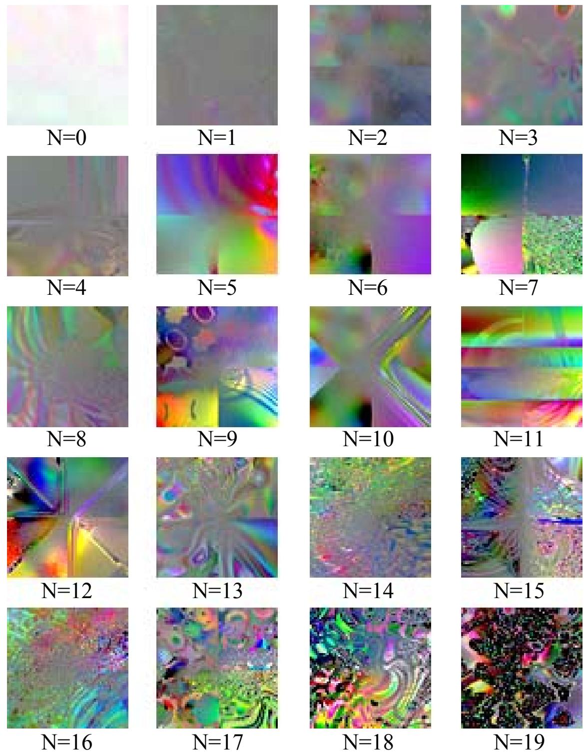


Figure 5.29: A small gallery of artworks produced by agents with different preferences for novelty (N) ranging from N=0 to N=19. In each case a 3-agent group were given the same prototype artwork to seed their evolutionary searches.

Another advantage of exploring design spaces using a tool as shown here is that there already exist many design tools that could be used by a curious design agent to explore new design domains with relatively little work. For example, a number of interactive evolutionary systems produce visual output that could be explored using the agents described here with only minor adjustments, e.g. Dawkins (1987), Sims (1991), Todd and Latham (1992), Baker (1998), Baker and Seltzer (1994; 1998), Graf and Banzhaf (1995), Coates (1997), Tabuada et al. (1998a; 1998b), Witbrock and Reilly (1999), and Lewis (2000).

Finally, curious design agents interact with the design tool using the same means as human users. This makes the possibility of produce curious design agents that explore design spaces in a way that human users can understand a real possibility and suggests that curious design agents could collaborate with designers in the exploration of a design space by sharing the same design tools.

5.4.6 Conclusions

The results of this experiment appear to confirm Martindale's hypothesis, at least for curious agents: the search for greater novelty produces more complex forms. The result that a linear relationship may exist between the preferred novelty of an agent and the fractal dimension of artworks produced suggests that the consequences of applying the Law of Novelty may be observed in cases where there is only a moderate pressure to find novelty: in these cases designs may be produced with a moderate rise in complexity over those produced with no preference for novelty. More research is required to determine whether this might be the case in general, and whether it applies in human creativity as well as curious exploration modelled here.

5.5 CONCLUSIONS

This chapter has presented experiments with curious design agents that have explored issues surrounding their use in computational models of creativity, in particular:

1) Some example implementations of curious design agents using the framework given in Chapter 3 have been demonstrated.
2) The implementations have demonstrated that curious design agents can be applied using three different modes of design interaction.
3) The experiments have provided examples of design spaces with different degrees of complexity explored by curious design agents.
4) The results have provided insights into the exploratory behaviour of curious design agents, e.g. the punctuated discovery of emergent shapes in Reflect-a-Sketch.
5) The results have also provided insights into the effects that curious behaviour has upon the long-term memories of curious design agents.
6) Analysis of the products of curious design agents has suggested a strong link between preferred novelty and the complexity of resulting works.

This chapter has demonstrated that curious design agents can explore design spaces using a number of different methods. The model of curiosity achieves this generality because it is concerned with the internal processes involved in learning and not the specifics of the design problem.

The methods described in this chapter offer different possibilities for developing future models of innovative and creative designing. The development of curious agents that directly manipulate design materials offers the possibility to develop detailed models of creative activity such as sketching. In these models, the functions implementing curiosity make up a relatively small part of the whole system, but they perform the important function of guiding the exploration of design possibilities within a reflective conversation with the design medium. Unfortunately, the number and complexity of the support systems needed to develop such an agent is formidable.

The second design method examined in this chapter, parametric configuration, reduces the number of complex support systems for a curious design agent, but at the cost of specialising it to the exploration of a domain using a specified set of parameters. This approach may be useful in design tasks where it is desirable for an agent to find innovative solutions within a specified design space. The example system shows that the representations that such an agent builds of the design space is different from those of an agent that explores the space without a bias to find novel designs. The "map" of the design space generated by a curious design agent highlights atypical designs and this may prove a useful tool when searching good places to look for innovative design solutions.

The curious design agents that use design tools show the most promise for developing sophisticated models of designing without spending time developing highly complex design agents. The decoupling of the design process from the more strategic guidance of the design process towards interesting design solutions allows curious design agents that use tools to be applied with relative ease to any domain that they can sense appropriately using design tools that they have the abilities to control. Interactive evolutionary design tools may provide an excellent vehicle for future curious design agent research as plenty of examples already exist of evolutionary design systems that could be controlled by a curious design agent that simply knows how to evaluate the interestingness of design products.

Chapter 6

Designing for Other Agents

This chapter extends the work presented in the previous chapter by presenting two novel applications of curious design agents that produce designs evaluated by other agents. The applications presented are autonomous design systems that require no human intervention for them to exhibit interesting design behaviour.

The aim of this chapter is to show that curious design agents can be used to study social creativity. The computational study of social creativity has been neglected because of the difficulties of developing complex design systems that can interact with real-world creative fields. The approach taken here is to use the ability of curious design agents to determine the potential interestingness of a design that was demonstrated in the previous chapter as the basis for social definitions of creativity valid within a closed simulation consisting of other curious design agents.

The first application presents an agent that explores a non-visual design space for interesting emergent behaviour in a crowd of pedestrian agents. The pedestrian agents act as consumers of the design agent's work, returning evaluations based on their individual experiences. The second application shows how multiple curious design agents can be used to develop models of social creativity within a peer group sharing works and evaluations.

6.1 DOORWAY DESIGN FOR EMERGENT CROWD BEHAVIOUR

Computer models of pedestrian movement have been used to provide valuable tools for designers when planning or modifying pedestrian areas in large buildings like railway stations or shopping malls (Major et al., 1998). The use of pedestrian agents in this work is not to simulate the details of human group behaviour, but to illustrate how the complex behaviour of consumers, in this case pedestrian agents, can have unexpected consequences on the design process.

6.1.1 Motivation

The applications of curious design agents presented in the previous chapter were all within visual domains. Visual emergence played an important role in each case as the agent curiously explored the space of possibilities. Emergent forms are unexpected and as such are sources of novelty and surprise.

The computational model of curiosity presented here has been developed in recognition of the fact that design emergence is more than just shape emergence: it models an interest in the emergence of unexpected group behaviour in crowds of simulated pedestrians. The task of the curious design agent is to explore a space of possible doorway designs that allow crowds of simulated pedestrians to pass in opposite directions.

While the problem of designing a doorway is conceptually simple, the complex interactions between the pedestrians mean that emergent group behaviours play a critical role in determining the performance of different designs. Therefore the initial statement of the design problem is necessarily ill defined: it cannot include a description of every relevant detail of emergent group behaviour in advance. This provides a similar problem to those faced by human designers: our design agent's task includes both problem finding and problem solving.

6.1.2 Implementation

The design agent used in these experiments is similar to the Spirograph explorer presented in Section 5.3; both agents use parametric configuration to explore design spaces. The main difference between the two agents is that the agents used here sense a much smaller number of variables; only two variables representing agent evaluations of doorways compared to the 1024 pixel values for each Spirograph pattern. As is often the case, presenting the design agent with a higher-level representation of a design problem dramatically reduces the complexity of the problem facing the agent; despite tackling the closest thing to a "real design problem" the curious design agent presented here is the simplest and has the easiest learning task.

6.1.2.1 Learning and Novelty Detection

The agent in this application uses two Habituated Self-Organizing Maps (HSOMs) to estimate the novelty of a situation (Marsland et al., 2000c). The first HSOM estimates the novelty of a design by categorizing a representation of the design solution. The second HSOM estimates the novelty of the performance of the design by categorizing a profile of the design situation that includes representations of the design solution, the design problem and an evaluation of the design's performance.

The inverse of the novelty detected by the first HSOM is used to estimate the familiarity of a design. The output of the second HSOM is used to estimate the novelty of the design performance. The novelty of a design situation is calculated as a product of the familiarity assigned by the first network and the novelty assigned by the second. Consequently, significant novelty is only detected when a familiar design has an unfamiliar performance.

6.1.2.2 Pedestrian Agents

A simple crowd management problem is used to illustrate the behaviour of our curious design agent. The problem is to design a doorway to facilitate the efficient and comfortable movement of crowds of pedestrians travelling in opposite directions. A pedestrian simulator was developed to evaluate doorway designs. A screenshot of the pedestrian simulator is shown in Figure 6.1.

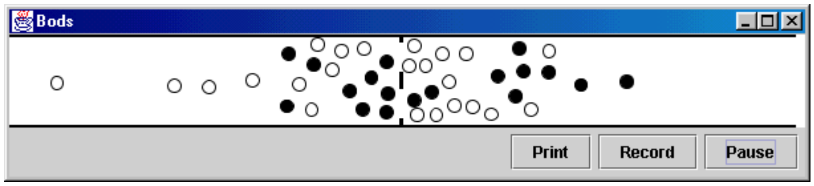


Figure 6.1: Screenshot of pedestrian simulator running a crowd simulation for a double doorway design. The black dots represent pedestrians moving from left to right; the white dots are pedestrians moving right to left.

6.1.2.3 The Social Force Model

The "social force model" is a microscopic model of pedestrian behaviour that simulates the behaviour of individual pedestrians to model self-organising phenomena in crowds (Helbing, 1991). Helbing and Molnár (1995) developed the social force model of pedestrian behaviour to simulate the pedestrian crowd movements to gain a better understanding of empirical results.

The "social forces" in the model do not represent forces exerted upon a pedestrian; rather they are an approximation of the internal motivations of the individuals to move in certain directions. Motivations include moving away from walls, keeping together with group members and moving towards goals.

Despite its simplicity, computer simulations have shown that the social force model is capable of realistically describing several interesting aspects of collective pedestrian behaviours observed in empirical studies (Helbing and Molnár, 1997). The social forces modelled in these experiments are listed in Table 6.1. Detailed mathematical descriptions of these forces can be found in Helbing and Molnár (1995).

	Description of social force
1.	Pedestrians are motivated to move as efficiently as possible to a destination.
2.	Pedestrians wish to maintain a comfortable distance from other pedestrians.
3.	Pedestrians wish to maintain a comfortable distance from obstacles like walls.
4.	Pedestrians may be attracted to other pedestrians (e.g. family) or objects (e.g. posters).

Table 6.1. The social forces modelled in the simulations of pedestrian crowds.

6.1.2.4 Evaluating Virtual Environments

Designs are evaluated using mathematical performance measures suggested by Helbing and Molnár (1997) that evaluate the efficiency and discomfort for each pedestrian agent moving through an environment. Efficiency is measured for a pedestrian as the average difference between the speed it is walking towards its goal and its desired walking speed. The efficiency of an environment, *E*, is calculated as shown in Equation 1.1, where *N* is the number of pedestrians α and the bar denotes a time average, $\vec{v}_\alpha$ denotes the velocity of an agent α, $\vec{e}_\alpha$ is a unit vector indicating its desired direction of travel and v_α^0 denotes its desired speed.

$$E = \frac{1}{N}\sum_{\alpha}\frac{\overline{\vec{v}_\alpha \cdot \vec{e}_\alpha}}{v_\alpha^0} \qquad (0 \le E \le 1) \tag{6.24}$$

Discomfort[7] is calculated as a function of the number of direction changes during a simulation that a pedestrian must perform in order to negotiate the built environment and other pedestrians. The discomfort measure, D, reflects the frequency and degree of sudden velocity changes, i.e. the level of discontinuity of walking due to necessary avoidance manoeuvres.

$$D = \frac{1}{N}\sum_{\alpha}\frac{\overline{\left(\vec{v}_\alpha - \overline{\vec{v}_\alpha}\right)^2}}{\left(\vec{v}_\alpha\right)} = \frac{1}{N}\sum_{\alpha}\left(1 - \frac{\overline{\vec{v}_\alpha}^{\,2}}{\overline{\left(\vec{v}_\alpha\right)^2}}\right) \qquad (0 \le D \le 1) \tag{6.25}$$

Hence, the optimal configuration of an environment regarding the pedestrian requirements is the one that maximizes its evaluations of efficiency and minimizes its evaluations of discomfort. Like an architect, the primary concern of the design agent used here is the "subjective experience" of the pedestrian agents visiting the environment it has designed for them. The emergence of self-organising behaviour at the doorway translates into different levels of efficiency and comfort reported by the pedestrian agents that may or may not have been predicted given experiences of similar designs.

It should be stressed that our curious design agent does not attempt to optimise its designs in the computational sense. Instead the design agent is motivated to explore the space of possible designs. It is equally motivated to investigate good and bad designs, e.g. inefficient designs can be interesting if their inefficiency is unexpected.

6.1.3 Experiment 1: Assessing the Novelty of a Two Door Design

To illustrate the judgement of interest by a curious design agent in different doorways, three designs for a doorway were created. The three doorway designs were for a narrow door, a wide door, and a combination of two narrow doors, as shown in Figure 6.2[8]. The aim of this experiment is to show how a curious design agent can identify the onset of interesting emergent crowd behaviour from the unexpected evaluations given by pedestrian agents.

6.1.3.1 Method

The doorway designs were tested using different numbers of pedestrians simultaneously trying to get through the doorway, crowds ranged in size from 1 to 51 pedestrians in increments of 10. The efficiency and discomfort measures from the simulations were combined into a single evaluation measure for each simulation. The best evaluations of three trials conducted at each crowd size are shown in Figure 6.3.

Unsurprisingly, all doorways performed equally well with only one pedestrian passing through it at a time. As the number of pedestrians increases the crowds display an oscillatory behaviour around doorways where one group of pedestrians gains control of the whole door at a time. The control of the doorway switches back-

[7] Helbing and Molnar (1997) refer to this performance measure as *uncomfortableness*.
[8] Videos of typical simulation runs with narrow, wide and double doors can be found on the accompanying CD-ROM: see Appendix B for details.

and-forth in direction as the numbers of pedestrians on either side of the doorway change.

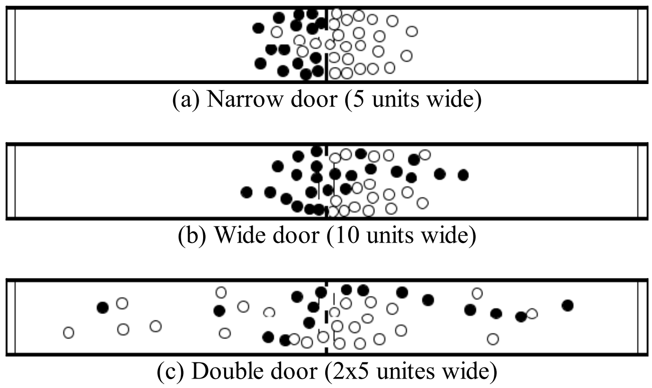


Figure 6.2: Screenshots of the simulations of pedestrian flow through (a) a narrow, (b) a wide, and (c) a double doorway design with a crowd of 40 pedestrians. The black circles indicate pedestrians travelling from left-to-right across the doorway and the white circles indicate pedestrians moving from right-to-left. The feint lines either side of each doorway indicate the targets that each agent walks towards to pass through the doorway.

The performance of the narrow doorway design quickly deteriorates to give consistently bad evaluations as the number of pedestrians increase. The wide doorway design maintains a very high performance for 11 pedestrians but its performance reduces dramatically, by almost 30%, as the number of pedestrians increases to 21. The performance of the wide door degrades more slowly over as the crowd sizes continue to increase from 31–51 pedestrians.

The performance of the double doorway design degrades even more slowly than the wide doorway design. For small crowds with less than 11 pedestrians the wide doorway design performs better but as the numbers of pedestrians increase the double doors outperform the wide door.

The double doorway design's superior performance in crowded conditions is a consequence of an emergent organisation. The two doors become specialised in the transfer of pedestrians moving in a single direction for relatively long periods of time. This can be seen in the double doorway simulation shown in Figure 6.2, pedestrians travelling from left to right pass through the top door while pedestrians travelling right to left pass through the bottom door.

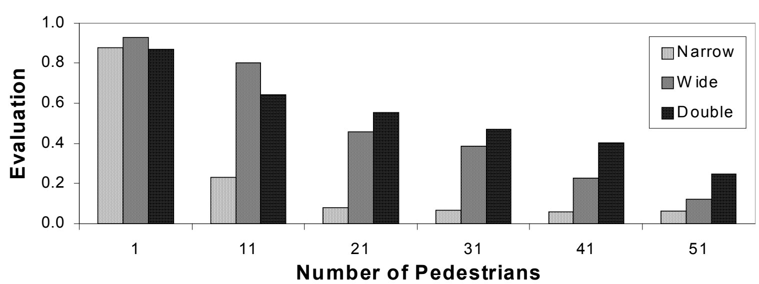


Figure 6.3: The best combined efficiency and discomfort evaluations for narrow, wide and double doorway designs for different crowd sizes (1–51 pedestrians).

6.1.3.2 Results

The evaluations of each doorway design were presented to a curious design agent in ascending order of pedestrian numbers. The evaluations of the narrow doorway were presented first, the wide doorway evaluations second and the evaluations of the double doorway were presented last. The best novelty measures of three trials are presented in Figure 6.4.

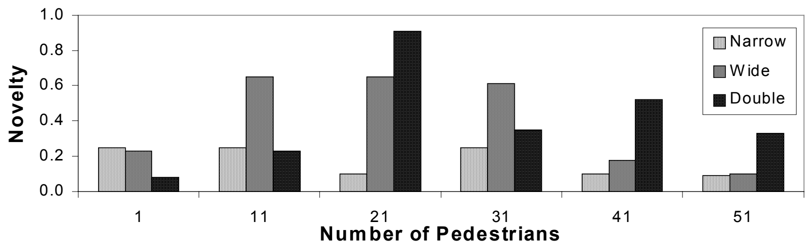


Figure 6.4: The greatest novelty detection for narrow, wide and double doorway designs for different crowd sizes (1–51 pedestrians).

Very little novelty was detected for the narrow doorway design at any crowd size. This is due to the lack of experiences against which the novelty detector could compare performances and the fact that the narrow doorway had consistently bad performance with more than one pedestrian.

The relatively high (~0.6) novelty measure for the wide doorway simulations with 11, 21 and 31 pedestrians indicate the improved performance of the wide doorway over the narrow doorway. The novelty of the wide doorway design drops at larger crowd sizes as the characteristics of the wide doorway are learned.

The novelty assessments of the double doorway design show very high novelty measures for simulations using 21 pedestrians, highlighting the resistance of the double doorway design to the fall in performance suffered by the wide door. The subsequent levels of novelty for simulations involving 31, 41 and 51 pedestrians reflect the relative differences in evaluations as the advantages of the double door design are maintained and the characteristics of the new design are learned.

The results of this experiment show that novelty detection can identify the most interesting designs without extensive reasoning by comparing the relative performance of different designs under similar conditions. The same novelty detector was used in the next experiment to implement models of interest and curiosity for an autonomous design agent.

6.1.4 Experiment 2: Curious Problem Finding and Problem Solving

In this experiment a curious design agent was given two conceptual spaces to explore: a problem space and a solution space. The aim of this experiment is to show that a curious design agent can maintain interest in a design task by switching between problem-solving and problem-finding.

6.1.4.1 Method

The solution space was defined by two variables: the number of doors making up the doorway and the combined width of doors. The problem space was defined by a single variable: the total number of pedestrians in the two crowds trying to get through the doorway. All other variables of the simulation remained constant.

6.1.4.2 Results

Figure 6.5 shows the novelty detected over the course of a design session. The design agent was initially given a narrow doorway as a solution to the problem of moving a single pedestrian. The novelty of exploring this design soon decreases as the agent learns to accurately predict the doorway's performance, the agent's interest level quickly falls below its boredom threshold and it begins to explore the problem and solution spaces for more interesting situations.

Figure 6.5 shows the design agent switching between searching the problem and solution spaces as interest in a particular problem or solution wanes. The chart shows the "tailing-off" of novelty values as the characteristics of situations are learned. It also shows how the detection of novelty extends the period that an agent spends searching a particular space, especially the exploration of the problem space for trials 17–37 and 67–82.

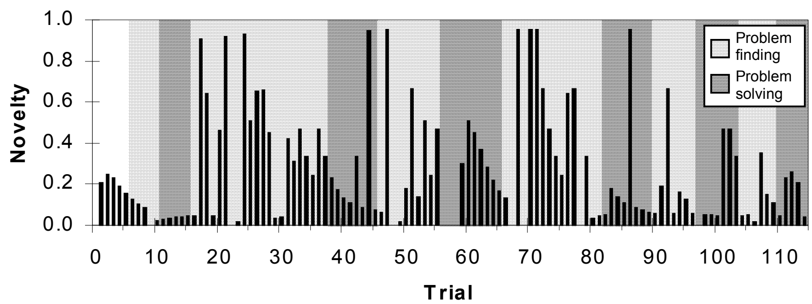


Figure 6.5: The results of using a curious design agent to explore the problem and solution spaces for doorway design. The chart shows the novelty detected for each simulation trial. The light shaded regions indicate that the design agent is problem finding and dark shaded regions indicate that the design agent is problem solving.

The highest peaks in detected novelty (~0.9) in the first half of the experiment (up to trial 68) all correspond to simulations using double doorway designs as these have significantly different characteristics to single doorway designs initially explored.

The high peaks in the second half of the chart correspond to simulations using wide doorway designs. This change in fixation occurs when the interest in double doorway designs subsides. In the second half of the design session the design agent is discovering an array of interesting situations where a wide door does not perform in the same way as a double door. At lower numbers of pedestrians the wide doorway does better than the double doorway, while at higher numbers of pedestrians it performs worse. Either way, the design agent finds situations involving wide doorway designs novel and maintains a higher level of interest in exploring this area of the design space than would otherwise be expected.

The change in fixation of the design agent from double to wide doorways illustrates a difference in exploration between a more conventional optimisation approach and one based on curiosity. The curious design agent did not explore the situations using wide door designs because they performed better than the double door designs. Instead, it explored the space of wide door designs because they did not perform as expected from previous experiences of the similar-yet-different double door designs. Exploring similar situations in this way allows a curious design agent to construct a better representation of the design space.

6.1.5 Discussion

Experiment 1 showed that novelty detection could be used to identify interesting situations where unexpected emergent properties play an important role in the evaluation of designs. Experiment 2 showed that a curious design agent can autonomously explore problem and solution spaces to identify interesting design situations from which to learn more about the design task.

6.1.5.1 Future Work

Possible extensions to this work include its application to more complex design tasks, possibly requiring a transition in design methods from parametric configuration to design tool-use. For example, a natural progression would be to apply curious design agents to the design of large public spaces like train stations where frequent interactions between pedestrians and the resulting emergent group behaviours have a significant impact on the performance of the space.

A different direction that this work could take would be the inclusion of "curious social forces" in the models of pedestrians. The motivation to explore new spaces for new experiences can be modelled in mobile agents as demonstrated by the curious agent exploring a 2D space demonstrated earlier, and as the work of Marsland et al. (2000a; 2000b; 2000c; 2001), Schmidhuber (1991a; 1991b; 1991c; 1997) and others have shown in other agent-based applications.

Modelling curiosity in crowds of pedestrian agents would permit the simulation of design problems where maintaining an interest in a space is as important as efficient movement or comfort. To illustrate this idea, consider the problem of designing an art gallery exhibition. The problem is one of maintaining interest for visitors to the gallery by exposing them to a sequence of similar-yet-different artworks. One might simulate this problem by substituting coloured patches for artworks and making the pedestrians interested in finding new colours.

Figure 6.6 illustrates the kinds of problems that might be modelled using curious pedestrians. The kinds of problems that the curious design agent would be concerned with exploring would include determining how best to keep an exhibition interesting throughout a pedestrian's journey around a gallery. To keep the experience interesting a gallery's design would have to take into account the preference of curious pedestrians to encounter similar-yet-different experiences.

The problems illustrated in Figure 6.6 are the formation of a crowd blocking passage at the gallery entrance and the streaming of pedestrians past paintings at the end of the gallery tour. The positioning of works around the walls causes both problems by producing "curious social forces" that steer pedestrians away from the desired direction of travel. The curious social forces are indicated on the gallery plan by light grey arrows. In the first case the painting at the end of the hallway is so different from those in the first room that the pedestrians prefer to remain with the familiar paintings causing a blockage. In the second case, the paintings in the last room are not visible from the previous room and so when they are encountered and they are discovered to be different from what was expected they gain little attention as the pedestrians stream by.

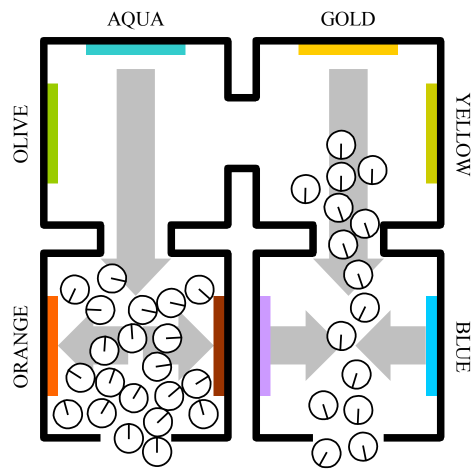


Figure 6.6: Crowding behaviour at start of gallery, streamed behaviour at end of gallery.

The importance of design problems such as the one presented above is that they show the potential for curious agents to play a role as consumers as well as designers. As consumers, curious agents have complex behaviour that changes over time with exposure. The problem of designing an interesting gallery is further compounded if one assumes that agents will visit the same gallery more than once. How does a curious design agent maintain the interest of visitors that have already experienced many of the works in previous visits?

6.1.6 Conclusions

The doorway design application shows one way that a design system can be constructed where design agents produce works that are evaluated by other agents, in this case pedestrian agents. More complex design problems could provide more opportunities to investigate the emergence of unexpected behaviour affecting evaluative judgements of pedestrians, however, a more interesting prospect is the incorporation of a model of curiosity in consumer agents, providing the opportunity to study evaluations that change with exposure to previous designs.

6.2 THE DIGITAL CLOCKWORK MUSE

The final application of curious design agents has been the most interesting. It demonstrates how the ability of curious design agents to assess the creativity of their products facilitates the production of complex models of social creativity.

6.2.1 Motivation

Gabora developed a memetic theory of social creativity that stresses the important relationship between innovation and imitation in the spread of creative ideas and cultural evolution (Gabora, 1996; 1997; 2000). Gabora has also developed a computational model, “Memes and Variations”, that demonstrates this theory for a fixed fitness function (Gabora, 1995). In “Memes and Variations” agents exchange information about ritual dance moves by imitating the movements of nearby agents. The success of any innovations made by the agents as a result of mutations are evaluated using an objective fitness function that calculates the number of correct limb positions over time. By using a combination of simulation, imitation and innovation Gabora showed how successful innovations quickly spread through a population of agents.

Colton et al. (2000) investigated a quite different type of culturally situated creativity in a study of agent based co-operative theory formation in pure mathematics. Colton et al. compared the performance of groups of collaborating agents with single agents. They discovered that small groups of collaborating agents with different search strategies outperformed single agents under a number of criteria. As part of their investigation, Colton et al. developed a definition of creativity appropriate to theory formation, based on the novelty of a theory's categorisation of a set of numbers. They used this measure of creativity to assess the relative performance of collaborating groups and found that larger groups with more diverse search strategies were more creative.

The computational model of creativity presented in this section uses a different approach to modelling creativity. Instead of using a fixed, objective definition of creativity it supports the emergence of socially defined notions of what and who are creative.

6.2.2 Implementation

The models of social creativity presented here have been developed using multiple agents of the type described in Section 5.4 with the addition of some communication skills. Each agent uses an interactive evolutionary algorithm to produce "genetic artworks". Agents can send and receive two types of messages; messages containing artworks encoded as Lisp expressions, and messages containing evaluations of previously sent artworks.

More interesting than the types of messages sent between agents are how the agent decides to send an artwork to other agents, and why an agent that receives an artwork would decide to pay for it. In both cases the decisions to take these actions are determined based on the interestingness of the artworks involved.

6.2.2.1 Communication Policies

During the evolutionary process an agent will determine the most interesting artwork in each generation and use that artwork as the parent of the next generation. If the interestingness of an artwork is so great as to breach a high threshold value then the artwork will be sent to other agents for evaluation. Interest ranges from 0.0 to 1.0, the interest threshold for communication used in these experiments has been fixed at 0.7 throughout.

Upon receiving an artwork an agent evaluates it according to its own experiences; experiences that will almost certainly differ from those of the originator. An artwork that was interesting for its creator may be boring to a second agent because it is too familiar or it may be uninteresting to a third agent because it is not familiar enough. Alternatively, an agent may find a received artwork more interesting than its own works, in which case it can use the received artwork as the starting point for a new evolutionary search, but before using an artwork received from elsewhere an agent must pay the creator of the interesting artwork some credit. The credit paid to use an artwork is proportional to the interest the receiving agent has in it. The amount of credit accumulated throughout a lifetime is used to assess how creative a particular individual is thought to be by other agents.

The receiving agent also has to add a record of the interesting artwork and the creating artist to a store of creative examples for posterity. Future generations of genetic artists can thus begin their search with artworks that were once considered creative. The record of interesting artworks can be used as a means to trace the development of artistic styles considered creative over time.

6.2.3 Experiment1: The Law of Novelty

The experiments described here extend the previous investigation of Martindale's hypothesis that the search for novelty is one of the primary motivators of creativity. It extends the study presented in Section 5.4 to consider the social consequences of applying the "Law of Novelty" (Martindale, 1990).

This experiment investigates the effects of applying The Law of Novelty on agents with different hedonic functions. The aim of this experiment was to show that agents are isolated if they fail to innovate in ways that other agents can appreciate.

In his thought experiment, Martindale only considered the case where an individual is ignored for repetition. The experiment presented here generalises the Law of Novelty by considering very similar works to be breaches of the law as much as exact replicas. Curious design agents generalise from their experiences, as a result they will ignore agents that not only repeat previous works, but also those that produce works that are very similar to previously experienced works.

This experiment also considers another extension to Martindale's law; it investigates the acceptance of agents that produce very novel works. Martindale was concerned with the promotion of novelty through social recognition, but the studies of preference judgements suggest that novelty is only appreciated in moderation as predicted by Berlyne's model of hedonic value. This experiment investigates the behaviour of a social group to an individual that produces radically novel works.

6.2.3.1 Method

The behaviour of agents with different preferences for novelty can be studied in a single simulation run. A group of agents was created for this experiment; most of the agents, numbered 0-9, shared the same hedonic function, i.e. the same preference for novelty (N=11). Two additional agents have different novelty preferences; one, "agent-10", has a preference for low amounts of novelty (N=3) and the other, "agent-11", has a preference for high amounts of novelty (N=19). The three hedonic functions used are illustrated in Figure 6.7.

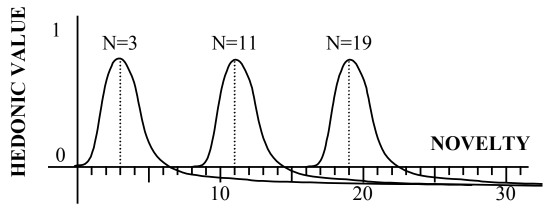


Figure 6.7: The hedonic functions for the agents used in the Law of Novelty experiment.

6.2.3.2 Results

The results of the simulation are presented in Table 6.2. The results show the agents with the same preference for novelty to be somewhat creative according to their peers, with an average attributed creativity of 5.57. However, neither agent 10 nor agent 11 received any credit for their artworks. Consequently none of the artworks produced by these agents were saved in the store of example artworks for future generations. When these agents expired nothing remained in the system of their efforts.

Agent ID	*Preferred Novelty*	*Attributed Creativity*
0	N=11	5.43
1	N=11	4.49
2	N=11	4.50
3	N=11	3.60
4	N=11	4.48
5	N=11	1.82
6	N=11	6.32
7	N=11	8.93
8	N=11	10.72
9	N=11	5.39
10	N=3	0.0
11	N=19	0.0

Table 6.2: The attributed creativity for a group of agents with different preferences for novelty.

Figure 6.8 shows how the network of communication links that has developed between agents that communicate artworks and evaluations on a regular basis excludes the two agents with different hedonic functions. In the screenshots of the running simulation the squares represent agents; the images in each square shows the currently selected genetic artwork for that agent, the number above each agent shows its attributed creativity, and the lines between agents indicate the number of rewarded communications between pairs of agents.

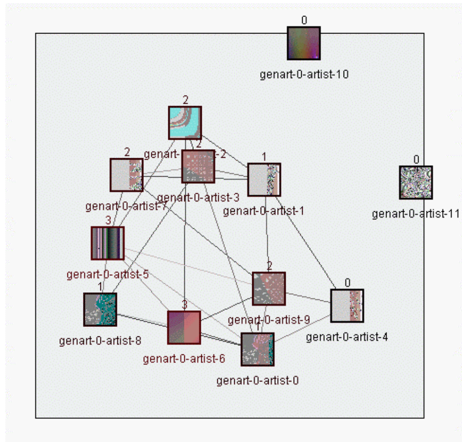


Figure 6.8: Screenshot of a running simulation of the Law of Novelty, shows graphically how agents that do not innovate appropriately are isolated. The agent named "genart-0-artist-10" prefers low amounts of novelty; "genart-0-artist-11" searches for high degrees of novelty.

To better understand the effects of an agent having a different hedonic function to the majority of agents in a population a series of similar simulation runs were performed where the difference between the majority preference for novelty and the two renegade agents is varied from 8, as in the current experiments giving N=3 and N=19, to 1, by giving the two agents hedonic functions favouring N=10 and N=12. The attributed creativity to the agents favouring high and low levels of novelty are shown in Figure 6.9. The figures plotted against the hedonic are the creativity attributed to an agent relative to the average creativity of the majority of agents that share the same hedonic function.

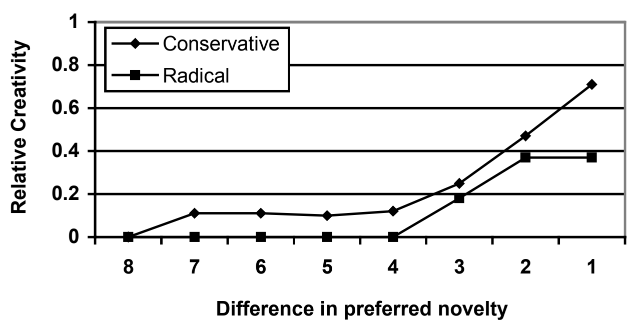


Figure 6.9: Relative creativity for a range of conservative and radical agents over a range of hedonic values that differ from the common hedonic preference (N=11) by the amount stated, i.e. conservative hedonic preference = 11 – hedonic difference, radical hedonic preference = 11 + hedonic difference.

Figure 6.9 shows that attributed creativity varies non-linearly with the difference between an agent's preference for novelty and the majority. It also shows a slight preference for the works of the conservative agent over the radical one.

6.2.3.3 Discussion

The results of this experiment appear to confirm Martindale's hypothesis generalises to the case where works that are very similar to ones previously experienced are ignored just as much as those that are exact replicas. To avoid being ignored an agent must produce some significant novelty that sets a work apart from previous examples. The results also show that while an agent must produce novelty to be considered creative, it must do so at a pace that matches its audience. There is no advantage in producing many highly novel works if the audience cannot appreciate them. In the first run of the experiment, the agent with a preference for high levels of novelty and hence rapid innovation was just as unsuccessful in gaining recognition as the agent with a low novelty threshold that did not innovate. Indeed, it appears from the series of experiments shown in Figure 6.9 that erring on the side of caution may be more beneficial that innovating too quickly but more work needs to be done to confirm this experimentally.

6.2.4 Experiment 2: Novelty Cliques

The previous experiment showed that agents become isolated when they innovate too slowly or too quickly because of low or high preferences for novelty. This experiment investigates the social behaviour of groups of agents with different hedonic functions. Some possible behaviours include: the production of comprised works that partly appeal to agents with different preferences for novelty, the production of works in bursts of high novelty followed by periods of low amounts of novelty, the formation of isolated groups that produce works for the other members of the group.

6.2.4.1 Method

A group of 10 agents were created for each run, half of them had a hedonic function that favoured novelty N=6 and the other five agents favoured novelty values close to N=15. The population of agents were allowed to evolve and communicate artworks for 50 time steps.

6.2.4.2 Results

Figure 6.10 shows the payments of creativity credit between the agents in recognition of interesting artworks sent by the agents. Two areas of frequent communication can be seen in the matrix of payment messages shown in Figure 6.10(a). The agents with the same hedonic function frequently send credit for interesting artworks amongst themselves but rarely send them to agents with a different hedonic function. There are a large number of credit messages between agents 0-4 and agents 5-9, but only one payment between the two groups – agent 4 credits agent 5 for a single interesting artwork.

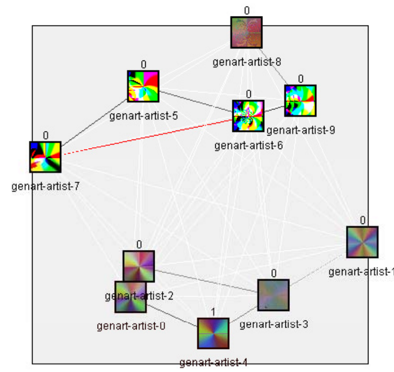
The result of putting collections of agents with different hedonic functions in single simulation appears to be the formation of cliques: groups of agents that communicate credit frequently amongst themselves but rarely acknowledge the creativity of agents outside the clique. As a consequence of the lack of communication between the groups the style of artworks produced by the two cliques also remains distinct.

Figure 6.10(b) is a screenshot of the running simulation that clearly shows the two cliques formed. The distances between agents are shortened when they communicate frequently. The different styles of the two groups can also be seen, with agents 0-4 producing smooth radial images with low a fractal dimension (~1.4) and agents 5-9 producing fractured images with clearly defined edges and a higher fractal dimension (~1.7).

Sender

Receiver	0	1	2	3	4	5	6	7	8	9
0		2	8	1	2					
1										
2	2	1		1	3					
3	4	5	2		5					
4	2	3	3	2						
5					1		6	1	3	5
6						3		4	5	1
7						3	5		1	4
8						4	3	2		4
9						1	4	4	4	

(a)



(b)

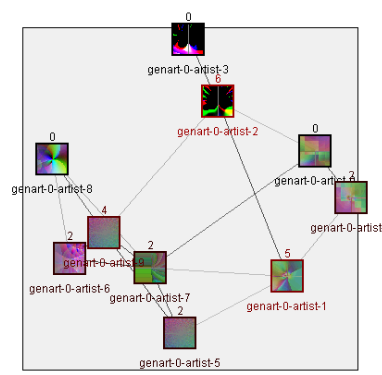
Figure 6.10: The communication of credit between two groups of agents having preference for novelty values N=6 and N=15. (a) A matrix showing the total number of messages carrying credit. (b) A screenshot of the running simulation.

A similar pair of groups was simulated with different hedonic functions that favoured N=9 and N=12. The communications of credit between agents is illustrated in Figure 6.11. The results show that while the cliques still form and communication of credit is still concentrated within these cliques, there are more inter-clique communications than before.

Sender

Receiver	0	1	2	3	4	5	6	7	8	9
0			1	2				2		
1			4	4				1		
2	1	1		1	2	2				1
3	3		1			1			1	
4	1	4		1				1		
5		1						1		2
6						1				
7						1	1		3	1
8								5		
9						1	2	1	2	

(a)



(b)

Figure 6.11: The communication of credit between two groups of agents having preference for novelty values N=9 and N=12. (a) A matrix showing the total number of messages carrying credit. (b) A screenshot of the running simulation.

An interesting observation about the nature of the communication between cliques can be made from looking at Figure 6.11(a) which shows that most of the payments between cliques came from the second group with preference for N=12; only one inter-clique payment was made by a member of the more conservative group that preferred N=9, i.e. between agent-1 and agent-5. This observation is consistent with the earlier observation that it is better to be too conservative than too radical when trying to gain the recognition of others with different preferences for novelty.

There are at least two possible explanations for this observation. The first is that agents with a higher preference for novelty can find the images produced by more conservative agents novel in comparison to the work of their fellow clique members. The second is that agents that prefer lower levels of novelty cannot appreciate the work of more radical agents and hence never attribute any credit to them. It is unclear from these results which explanation is more likely as either would explain the data. Further work may find that both behaviours play a role in the formation of cliques and the unequal communication of credit between them.

6.2.4.3 Discussion

The results of this experiment show that when a population of agents contains subgroups with different hedonic functions, the agents in those subgroups form cliques. The agents within a clique communicate credit frequently amongst themselves but rarely to outsiders. The stability of these cliques depends upon how similar the individuals in different subgroups are and how often the agents in one subgroup are exposed to the artworks of another subgroup. Further research is needed to determine whether other factors that can affect judgements of interestingness can similarly affect the social structure.

6.2.5 Future Work

The artificial creativity framework implemented here provides several opportunities for developing future models of social creativity. Three possible directions for future work are: (1) the simulation of larger creative societies, (2) the development of new types of agents, and (3) the development of more complex social interactions.

6.2.5.1 Large Creative Societies

The ability to simulate larger creative societies will permit the study of the spread of innovations (Gabora, 1997; Goldenberg et al., 2000) and styles. It may also facilitate the emergence of new fields as cliques attain a critical size. Spatial and topological relationships will become more important issues in large population models.

6.2.5.2 New Types of Agents

There are several other important players in creativity societies besides the producers of innovations including, e.g. consumers, distributors, critics, etc. Each has their own role to play in artificially creative societies; consumers evaluate products, distributors distribute products widely, and critics distribute their evaluations widely. These roles are illustrated in Figure 6.12.

Convincing other people that you’ve had a creative idea is often harder than having the idea in the first place (Csikszentmihalyi, 1999). In non-homogenous societies of agents, the selection of which agents to communicate with becomes important for agents seeking recognition from their peers.

6.2.5.3 Strategic Knowledge

Simulations of technological innovation in industry show that the consideration of the costs of innovation in decision-making can lead to complex behaviour (Haag and Liedl, 2001). Simulating similar costs in the design process may provide a better understanding of the economics of creative design in creative societies and the strategies needed to manage creativity with limited resources.

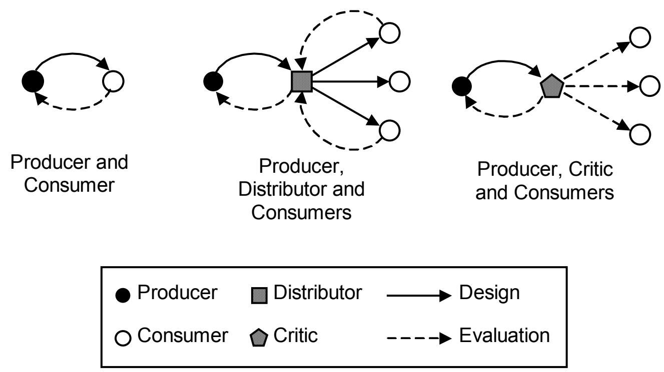


Figure 6.12: Three different types of individuals and their roles in the communication of designs and evaluations in creative design societies.

6.2.5.4 Dynamic and Distributed Domains

Providing a more dynamic model of the domain will complement the already dynamic model of creative societies presented in The Digital Clockwork Muse. Future implementations on The Digital Clockwork should attempt to capture the emergent nature of the domain observed in human cultures. Currently, the domain is implemented as a very simple database of possible starting points for agents to use to search for interesting designs: the domain contains all artefacts added to it during the course of a simulation run and the domain interaction policies for individuals are limited to simple storage and retrieval commands; there is no support for the revision of a domain's content over time. The static nature of the domains implemented so far has prevented an investigation into the honorific nature of Boden's H-creativity in artificial creativity systems. Expanding the range of domain interactions to include the revision of the domains content will allow such experiments to be conducted.

Beyond the expansion of the role of the domain as a database, there is also the possibility for distributing some, or all, of the domain's functions amongst the members of its associated field. The simultaneous existence of multiple views of the current state of cultural knowledge among the individuals in a field will allow the effects of incomplete and erroneous domain knowledge to be studied, providing a far more complex cultural environment for the individuals to explore and learn about as they traverse between culturally distinct niches.

6.2.6 Conclusions

The Digital Clockwork Muse project has demonstrated the utility of the artificial creativity approach studying social creativity in multi-agent simulations. The experiments have shown the emergence of definitions about whom and what are creative as a consequence of the search for novelty and the communication of works between individuals. The first experiment investigated the role that an agent's hedonic function plays in the recognition of its socially defined creativity. The second experiment showed that groups form when agents have different hedonic functions that favour different types of artworks.

In the above simulations the consensus of what is creative, i.e. those artworks that are stored as creative examples, has been demonstrated to be a function of both the individual's drive for novelty and the collective experience of the group of agents. The definition of a creative artwork is thus a social construct of more than one agent. The assignment of creativity to an agent is also an honorary term given to agents that consistently produced artworks appreciated by other agents.

6.3 CONCLUSIONS

This chapter has present experiments with curious design agents that have explored issues surrounding their use in closed-world simulations of creativity. These issues have included:

1) Curious design agents have been used to develop closed-world simulations of socially situated creativity involving designer and consumer agents with different views of the simulated world.
2) The behaviour of curious design agent able to switch between problem-finding and problem-solving has been examined.
3) It has been shown that it is possible to implement artificial creativity models using a heterogeneous population of curious design agents.
4) Emergent properties of an artificial creativity model, The Digital Clockwork Muse, have been investigated showing the emergence of notions of who and what are creative and the formation of emergent social structures.

This chapter has presented two approaches to developing closed-world simulations of socially situated creative behaviour. In the first application a curious design agent was designed for the satisfaction of the needs of pedestrian agents to pass efficiently through a space and maintain a level of comfort while doing so. The integration of more complex needs could make this type of system a useful tool for the study of designer responses to consumer demands that change over time. In particular, the integration of curiosity into the models of consumers would provide an opportunity to study the demand for creative designing as well as its satisfaction.

The second application presented an abstract model of social creativity within a community of designers (or artists). The details of the design artefacts are of secondary importance in the analysis of the social behaviour of designers although they play an important role in the design process. More important contributions to our understanding of social creativity can be gained from studying the emergent social behaviour observed in groups of agents, particularly the social structures formed by networks of communication between agents. The possibility for developing more complex social simulations is discussed in the next chapter.

Chapter 7

Future Agents of Curiosity

The applications presented in this thesis have demonstrated curious design agents applied in a variety of ways to a range of design domains. Each application presents unique possibilities for future work; some of these possibilities have been discussed in the application-specific discussion sections.

This chapter presents a discussion of the limitations of current curious design agents and some directions for future work. It also discusses how the model of curiosity developed in this thesis could assist an agent pass a Turing Test of creativity.

7.1 CURIOUS DESIGN AGENTS

This section seeks to address some reasonable criticisms of the curious design agents presented in this thesis. The intention is not to refute these criticisms but to use them as pointers to future work. Among the most important criticisms of the curious design agents presented are:

- *The curious design agents did not create anything new.* The curious design agents only ever explored micro-domains; their works have no value in human society and so the agents cannot be considered creative. If the test of a machine's creativity is that it produces works judged to be creative by human society then no curious design agent has succeeded in being the least bit creative.
- *Will the model of curiosity scale up to real-world problems?* AI models often show great promise in micro-domains that fails to materialise when applied to more complex problems. If the model of curiosity cannot scale to design problems of reasonable complexity this would suggest a problem with the approach.

- *The curious design agents did not have design goals.* Designers work to satisfy the needs of clients by setting design goals to achieve some level of usefulness. The curious design agents developed so far have no explicit design goals; this diminishes their claim to model designing or any other form of creativity requiring significant utility in the final product.
- *Curiosity is more than just searching for novelty.* The curious design agents model the curiosity as diversive exploration when "bored" to increase stimulation but Berlyne defines curiosity to also include specific exploration where an object is focussed upon to reduce stimulation.

7.1.1 The Problem with Micro-Domains

The use of micro-domains has a long history in Artificial Intelligence. One problem with the use of micro-domains in this research is that they preclude the possibility of curious design agents discovering anything of particular value to human society, but for the purposes of demonstrating the validity of the approach this was not considered a serious problem. Instead, the aim has been to show that a model of curiosity offers advantages in the development of autonomous design agents.

Knowledge-based design systems require large amounts of information to proceed and suffer from an inability to generalise that restricts them to narrow domains of expertise. In contrast, curious design agents require very little knowledge about a domain before they can explore its design space. Tool-using curious design agents only require the necessary abilities to operate the design tool and perceive the results. In some cases this may mean that a curious design agent could be assigned to new domains without any changes.

Figure 7.1 attempts to illustrate the difference in the level of competence and robustness between knowledge-based design systems and curious design agents. The width of a strip indicates the amount of autonomy that each type of design system has as an indication of the amount of competence that a system can gain through experience. Knowledge-based design systems have a high degree of competence in a narrowly defined domain, while the curious design agents developed so far have a much lower level of competence but can potentially handle a wider range of situations because they use a general-purpose heuristic to guide design space exploration.

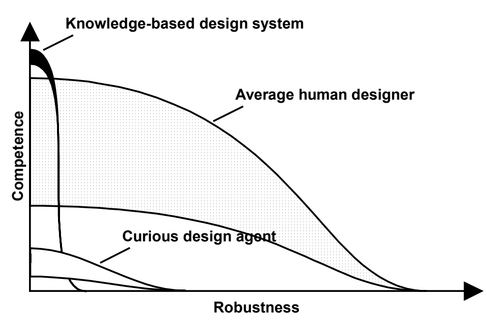


Figure 7.1: Comparing the competence and robustness of curious design agents with knowledge-based design systems and human designers (after Peters, 2000).

7.1.2 Scaling Up Curiosity

A more serious problem with the use of micro-domains is the quite reasonable concern that the model of curiosity might not scale up to allow curious design agents to tackle more complex design problems. Fortunately, the model of curiosity presented in Chapter 3 is based on the comparison of conceptual state variables and not on the comparison of design state variables. This means that as long as the agent has the necessary functions to construct a conceptual state describing its current design situation then the current model of curiosity will scale with the design problem.

The adaptation of conceptors and long-term memory in curious design agents is of more concern but there are plenty of options available to curious agent developers. Several different neural network architectures have been proposed as possible novelty detectors based on different technologies, e.g. feed-forward networks (Kohonen, 1993), combinations of multiple feed-forward networks (Schmidhuber, 1991a) and adaptations of SOMs (Marsland et al., 2000).

Marsland et al. are currently developing the Grow When Required (GWR) network architecture, a form of growing self-organising network, for the purposes of supporting novelty detection in a curious agent (Marsland, 2001). Growing self-organising networks should prove useful for implementing models of curiosity because they combine the strengths of SOMs and ART networks (Fritzke, 1996). Recurrent neural networks, e.g. Pollack's RAAM or Elman's SRN (Pollack, 1990; Elman, 1990), should also be useful for encoding temporal sequences in the conceptual state of an agent for the detection of surprise.

Curious design agents are not limited to using novelty detectors based on neural networks. For example, the curious experience selection mechanisms of Scott and Markovitch (1989a; 1989b) used a novelty detection mechanism based on Shannon's measure of entropy. Baker et al. (1999) developed a scheme for novelty detection in text documents based on a hierarchical classification scheme that they used to track breaking news stories.

Gomes et al. (1999) presented a function for novelty evaluation in design for use with case-based reasoning (CBR) design systems. Comparisons of the novelty evaluation with the judgements of a design expert showed that it was reasonably accurate. Macedo and Cardosa (2001a, 2001b) took the next logical step and developed a model of surprise and curiosity within a CBR design system as both a search heuristic and a model of emotion. CBR is a popular approach to modelling design reasoning and Macedo and Cardosa's approach to modelling curiosity shows great promise as a scalable agent framework.

7.1.2.1 Adaptive Novelty Detection

Although the current model of curiosity should scale it is not particularly sophisticated. Curious agents with simple models of curiosity can become trapped in non-deterministic areas of an environment because the non-deterministic nature of the environment generates a continuous stream of novel experiences that make it perpetually interesting. Schmidhuber (1991a) demonstrated a simple solution to the problem by using two predictors to detector novelty rather than one. Schmidhuber's model of adaptive curiosity uses one predictor to detect novelty in the environment and a second predictor to determine the reliability of the first predictor. If the first predictor is unreliable within a certain part of the environment then the novelty detected by it will be high, however, the second predictor will be able to model this unreliability and so discount the novelty detected by the first predictor.

Peter's (2000) hierarchy of attention systems based on novelty detectors models a similar approach to adaptively attending to novelty in the environment. Using an adaptive novelty detector should allow curious design agents to explore complex design spaces more robustly and avoid fixating upon unpredictable aspects of the design problem.

7.1.3 Creative Designing Requires Goals

None of the curious design agents presented in this thesis model an interest in anything other than novelty. Obviously, this does not model the interest that people show in a situation when they have a set goal as in problem-solving. Models of interest in future agents must include other important features of the design problem. Design agents must model an interest in satisfying the needs of consumers. For example, future doorway design agents should not only explore the interesting self-organising behaviour of pedestrian agents but also have an interest in maximising efficiency and minimising discomfort.

The most interesting aspect of expanding the scope of interest in curious design agents is determining how to assign importance to the requirements of the design product to be useful vs. the motivation to find a novel solution. In other words, how to dynamically determine the importance of importance (Waltz, 1998).

The doorway design application suggests one approach to assigning importance. In that application the level of interest was used to determine when to switch between exploring solution and problem spaces based on the assumption that when an agent's interest level falls below a boredom threshold there is little more for it to learn while exploring the current space. Using the same assumption, a curious design agent might control the transition between conceptual and detailed designing by adjusting the relative importance of novelty vs. utility judgements in determining interest. The advantage of this approach is that it provides a smooth transition from conceptual to detailed design and allows the retreat back into conceptual design if no useful designs can be found within a period of detailed designing.

7.1.4 Curious Exploration

The curious design agents presented here have modelled diversive exploration for the most part, i.e. the exploration of a space in search of stimulation. These agents did not need to model specific exploration, i.e. the exploration of a space to reduce stimulation, because they had learning systems able to learn quickly within the simple domains presented. Agents exploring more complex design spaces will need to be able to conduct specific exploration to ensure that they have properly learned about an object of curiosity before continuing to explore.

7.2 FUTURE AGENTS OF CURIOSITY

Agents have been applied to situations where users face information overload, in particular, in the filtering of information streams such as Internet newsgroups (Maes, 1994). Technologies similar to curious design agents may play an important role in future CAD systems by reducing the amount of information presented to an architect throughout the design process.

7.2.1.1 Reducing Information Overload

Architects increasingly face the problem of "information overload" as they try to explore complex design spaces for innovative solutions. Applications of novelty detection in data-mining, text-retrieval and information filtering (Baker et al., 1999; Maes, 1994) suggest that curious design assistants could be used to find interesting information in the preparatory stages of design when a designer must gather source material to inform future design decisions.

7.2.1.2 Generating Interesting Solutions

Although generative design systems can assist designers by producing many possible design solutions, these systems often suffer the same problem as the models of creativity discussed in Chapter 2; they cannot distinguish potentially interesting designs from those that are not because they have no model of how unexpected a particular design is.

Curious design assistants present two possible solutions to this problem. The first solution is to filter the designs presented by a generative design system for interestingness before presenting them to a user. The second solution is to have a curious design agent autonomously explore a design space using a generative design system and then prepare a report on potentially interesting solutions at the end of the exploration. In either case, curious design assistants will have to be shown to model a user's preferences.

7.2.1.3 Modelling User Preferences

Baluja et al. (1994) attempted to model user preferences using neural networks trained on a user's evaluations of genetic artworks. The user evolved genetic artworks using an interactive evolutionary system, similar to the ones used by curious design agents in Chapters 5 & 6. The aim of the study was to train neural networks to evaluate artworks similarly to the user, based on features within each genetic artwork.

The researchers attempted several different types of networks ranging in complexity but found that for the most part the networks could only predict which images were likely to be uninteresting with any accuracy. It was concluded that the disappointing performance of the networks was due to a lack of sophistication in the image processing and learning systems, however, as Baluja et al. observed: "users often will choose an image because it is different than the other images on the screen." This simple observation suggests that a system based on novelty detection and curious selection might better predict the preferences of a user.

7.2.1.4 A Curious Design Assistant

As a proof-of-concept, a system has been developed that allows a curious design agent and a person to use the same tool to evolve 2D structures[9]. The structures evolved by the agent are called “horns”, name after the similar 3D structures evolved by Latham (Todd and Latham, 1992). Horns are constructed by applying a sequence of morphological processes to simple graphical elements to produce complex structures[10]. The curious agent uses an ART-based novelty detector to determine the novelty and a linear hedonic function to favour the most novel horn in each generation with respect to its experiences of other horns within an evolutionary run.

A preliminary study compared an agent’s interestingness rankings with a user’s selections. The results in Figure 7.2 shows that the curious design agent could predict the most interesting structure in a population (i.e. assign it a rank of 1) with up to 50% accuracy; taking the top 3 rankings as likely candidates for selection improves this score to between 60% and 72% accuracy. Unlike Baluja et al.’s study, the agent in this system is not designed to learn a user’s preference, rather it is a model of user preference based on the empirical findings. The results of the initial study suggest that up to 72% of selections can be explained as a preference for novelty.

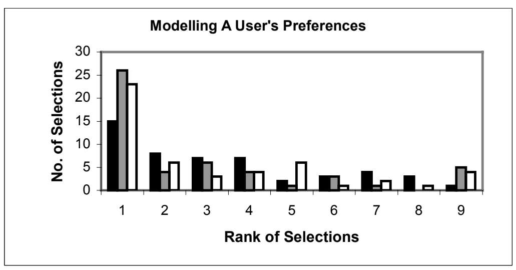


Figure 7.2: Results of pilot study to model a user's preferences using a curious design agent. Chart shows the number of selections made by a user against the preference judgements of a curious design agent ranked from 1–9, where a rank of 1 is given to the most interesting structures. Three trials of 50 selections in each trial are shown.

The system has the ability to function as an “auto-pilot”, guiding the evolution of new horns along the most interesting paths that present themselves. Figure 7.3 illustrates the selections of the curious design agent for a short run of 30 generations. The figure shows the horn selected at each generation and below each selection the population of nine horns that it had to choose from.

---

[9] This system is available on the accompanying CD-ROM: see Appendix B for details.
[10] The implementation of horn evolution used in this work is based on code developed by Marius Watz, that is freely available on the Internet at http://www.notam.uio.no/~mariusw/form/java/horn.html.

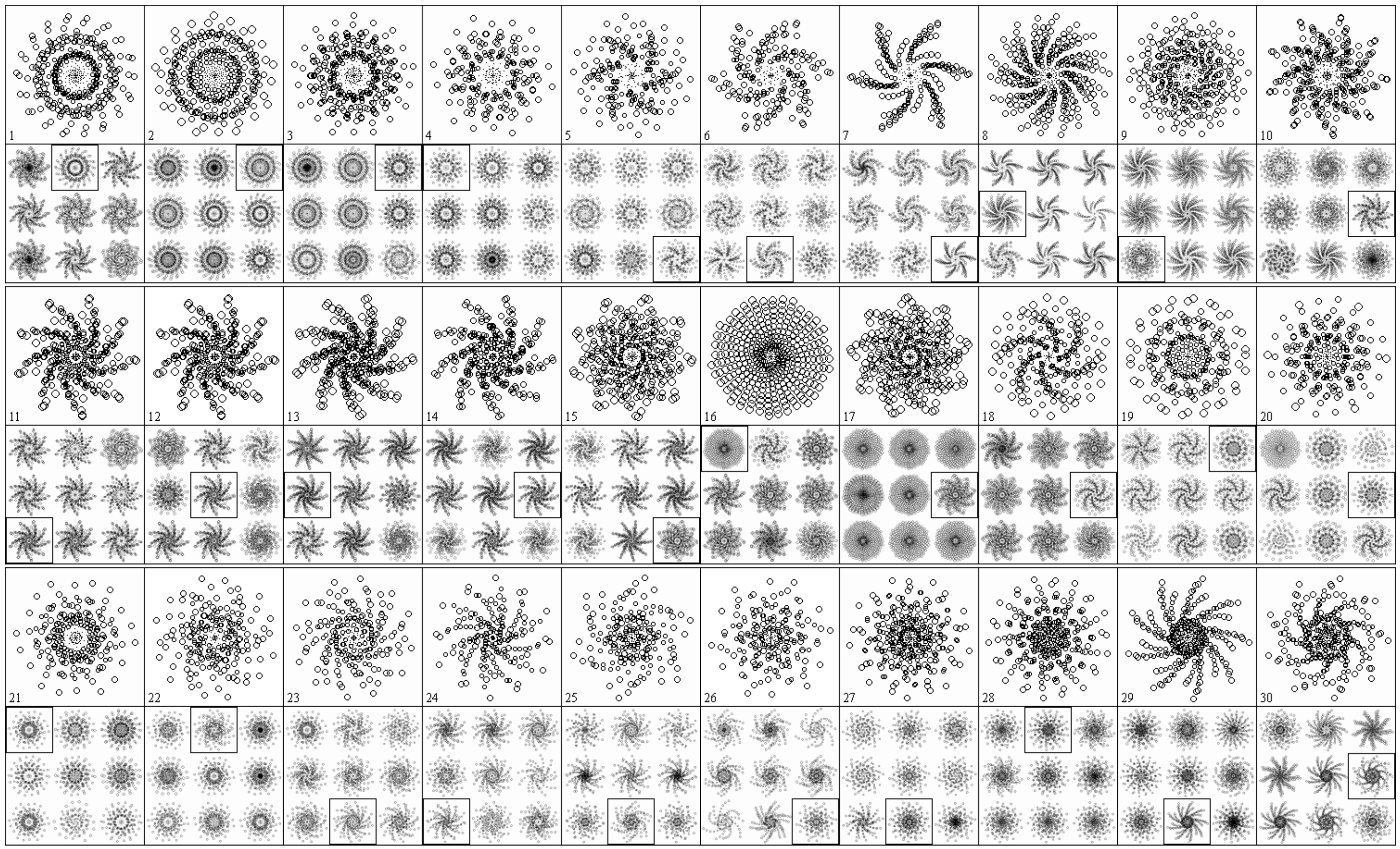


Figure 7.3: The selections of the curious design agent during a run of the interactive horn evolver. Each selection is illustrated with a large image of the horn selected above an image of the population from which it was selected with a box is drawn around the horn selected.

7.2.1.5 Finding Interesting Problems

The doorway design example showed that a curious design agent could find interesting design problems using the same processes that it used to find interesting design solutions. In the same way that curious design assistants can be used to generate interesting design solutions they could also be used to generate interesting design problems thereby assisting the designer identify potential problems early in the design process. Also, because curious design agents are interested in all atypical behaviour, curious problem-finding assistants could also be used to identify unexpected opportunities in design situations, as in the example of the spontaneous self-organisation of crowd behaviour through a double doorway presented earlier.

7.3 RECONSIDERING ARTIFICIAL CREATIVITY

This thesis began by considering the question of age-old question of whether computers could ever be creative. The need for a computer to recognise the novelty of its works was identified as a necessary requirement if computers are ever to be considered creative; for it is the ability to recognise novelty that is lacking in existing models of creativity that can produce many works but cannot differentiate between mundane and creative solutions. The intervening chapters have shown how the ability to recognise novelty and act upon that recognition to promote the further production of novel works can be added to existing agent models and how it can implemented using existing technologies.

The curious design agents presented earlier represent a new vehicle for the computational study of creativity. The applications presented in Chapters 5 & 6 have demonstrated the possibility of more completely modelling creative behaviour by using curious design agents to model both personal and social creativity. The aim of this section is to develop a better understanding of the implications of the behaviour of curious design agents for the computational creativity debate.

This section re-examines two commonly proposed tests for artificial creativity and then re-visits Turing's famous test for intelligent behaviour with the intention of identifying the core question that creative systems should be able to answer if they are to prove themselves creative.

7.3.1 Testing Machine Creativity

How can artificial creativity be tested? There are commonly considered to be two aspects of a system that can be tested for creativity; firstly, the products of a system can be tested for novelty and usefulness, and secondly, the processes of a system can be tested for similarity to recognised creative processes.

The problem with testing the products of a computational system is that it is too readily passed by systems that do not claim to model creativity. The generation of novel and useful works does not require that the generative system model creativity if a human observer can do the evaluation of novelty. This has been the case with many of the systems that have produced innovations (see Section 2.2 for some examples).

As noted by Boden (1990), how a machine came to produce a creative work appears to be an important criterion for many people. For example, Cohen refuses to attribute his program AARON with creativity, even though it has produced many works in its unique style that have been displayed in galleries alongside the work of human artists. Cohen does not attribute AARON with creativity because he does not consider AARON to be an implementation of suitably general creative processes (Cohen, 1999). The need for candidate creativity systems to implement processes regarded to be at the core of human creativity, e.g. analogy-making, seems to be rooted in the well-founded suspicion that a seemingly creative system could be constructed by simply combining a well-chosen "bag of tricks" that give the illusion of creative behaviour within a narrow domain; however, it should be noted that the same people who may discount computational systems this way are generally prepared to accept, without access to internal mental processes, the creativity of other people.

One problem with the relying on a test of whether or not a system is using a creative process is that using a creative process does not guarantee that a system will generate creative products. In addition, it is unclear what constitutes a creative process; although some rational thought processes have been identified as being useful in generating creative ideas, e.g. analogy-making, there are still many researchers that believe that creative thinking is conducted mainly as a series of sub-conscious processes, about which little is known.

Bringsjord (2001) has recently proposed a test for the creativity of computer systems that requires that a person with complete knowledge of the internal workings of a computer system be unable to explain the production of a work within a generous period of time, e.g. a year. Bringsjord's test seems unduly biased against the computer, by allowing access well beyond what is possible with human subjects. The test almost seems to invoke the ancient meaning of inspiration, requiring a mythical breath of creativity to enter the machine, as it appears that any computer that could pass this test must go beyond the bounds of what is computationally possible. The more mundane alternative is that the processes involved are so complex that it would take longer to understand than is available. Creative processes may well be complex but there is no reason to assume so in advance: evolutionary systems are computationally quite simple and yet they are capable of producing novel and useful products.

7.3.2 Testing for Creative Behaviour

The development of agents that can assess the novelty of their own works and act upon that assessment to further explore design spaces suggests a third aspect of a computational system that can be tested; the behaviour of the system over extended periods of time. This test for behavioural creativity recalls the "imitation game" proposed by Turing as a test for intelligence. Turing's "imitation game" allows an interrogator to assess the behaviour of two participants using a series of probing questions and answers. If at the end of a session the interrogator cannot determine which of the participants is a computer then Turing suggested that the machine must be declared intelligent.

It is tempting to assume that the evaluation of products constitutes a Turing Test for creativity, but this is not the case. Products alone do not provide an opportunity to probe further into the motivations and goals of the designer. The strength of the imitation game proposed by Turing lies in the open-ended nature of the possible dialog between the human interrogator and the participants of the test. As Turing (1950) notes, “the question and answer method seems to be suitable for introducing almost any one of the fields of human endeavour that we wish to include.” Turing gave the following extract of a possible dialog to illustrate his point:

> Q: Please write me a sonnet on the subject of the Forth Bridge.
> A: Count me out on this one. I never could write poetry.
>
> Q: Add 34957 to 70764
> A: (Pause about 30 seconds and then give as answer) 105621.
>
> Q: Do you play chess?
> A: Yes.
>
> Q: I have K at my K1, and no other pieces. You have only K at K6 and R at R1. It is your move. What do you play?
> A: (After a pause of 15 seconds) R-R8 mate.

In support of the use of the “imitation game” as a test of creativity, Hofstadter (1995b) argued that “*covert mechanisms* can be deeply probed and eventually revealed merely by means of watching *overt behaviour*” and that this “lies at the very heart of modern science”. Certainly, when Turing proposed his test he meant for it to encompass tests for creative behaviour as illustrated by the request to compose a sonnet. Hofstadter suggests that even in a scaled-down version of the Turing Test, the systematic probing of a system should reveal the nature of its creative abilities. For example, a computer system that relied on a narrowly defined grammar to produce works of a particular style would soon be uncovered by the formulaic nature of its responses to repeated requests.

7.3.2.1 The Importance of Asking “Why?”

One of the most important devices available to an interrogator in a Turing Test designed to ascertain the creativity of a computer system is the simple question “Why?” The use of this question is the key to uncovering the mechanisms at work when creative artefacts are produced as well as the context within which they should be considered creative. Asking a computer system to explain why its work is creative is also in agreement with Csikszentmihalyi’s observation that convincing others of the value of a creative work is often harder than producing it (Csikszentmihalyi, 1999).

The request for an explanation of why something should be considered creative is the logical extension of requirement that an agent be able to recognise that it has produced something creative. The response should reveal the context within which the computer system has judged its work to be creative. This is an important aspect of creativity that cannot be determined from either the product or the process tests of creativity described above. The situation that a creator was in when a work was produced is fundamental to any attribution of creativity. Producing the same work twice does not generally mean that a person is twice as creative; the requirement for novelty in creative works means that the second production is unlikely to be considered creative, no matter how creative the first work was determined to be.

Could future curious design agents pass a Turing Test of creativity? The Turing Test sets a very high standard for artificial creativity, how could a curious design agent hope to pass such a test? The important thing to note is that the curious exploration process relies upon the comparison of situations and this gives curious design agents the contextual understanding of the creative process needed to explain the creativity of their works. Consider the following dialog between an interrogator and a curious design agent set the now familiar task of designing a doorway:

> Q: Please recommend a design for a doorway to facilitate the efficient and comfortable passage of opposing crowds within a confined space.
> A: (After a suitably long pause.) I recommend that you use a double doorway design; use two standard width doors placed 0.25m apart in the centre of the wall.
>
> Q: Do you think your design is creative?
> A: Yes.
>
> Q: Why is it creative?
> A: The double doorway costs the same to install as a wide doorway measuring twice the width of a standard door but the double doorway is much more efficient at allowing crowds to pass through, and the crowds pass through with greater comfort. The efficiency of the double doorway was unexpected given its similarity to the wide doorway.
>
> Q: Why does the double doorway work so well?
> A: I don't know.

This question and answer session illustrates two important aspects of the representations built by a curious design agent (1) it can answer the question of whether it thinks it has produced a creative product, and (2) it can explain, to a limited extent, why it thinks it is creative. The doorway design knowledge needed to answer these questions is little more than what is already available to the doorway design agent at the end of a curious exploration of the space. The current doorway design agent can recognise the novelty, and hence potential creativity, of the double doorway design and it does so by relating its superior performance to the similar wide doorway design.

The last question in the above session also shows the limits of the model of curiosity currently implemented in curious design agents; although they can recognise and explore the novelty of a situation, they do not have the capacity to formulate ways of understanding the causes of novelty. How to provide an agent with the ability to explore aspects of a curious situation with such flexibility remains an open question.

7.3.3 Studying Creative Behaviour

The emergence of social behaviour, e.g. The Law of Novelty, and dynamic social structures, e.g. cliques, in The Digital Clockwork Muse suggests that the artificial creativity approach to developing models of creative societies may contribute new insights into the nature of creative design in socio-cultural situations.

Figure 7.4 illustrates the different levels at which creativity may be studied as a pyramid of emergent properties. Each level represents a different aspect of creativity that is emergent from the ones below it. The foundations of the creative pyramid are the processes internal to the creative agent that allows it to generate-and-test ideas. The result of executing these processes is the creative products. Traditionally, computational research has concentrated on these two levels by encoding processes thought to be important in creativity in a piece of software and getting experts to examine the results of running those processes to determine whether the processes are creative; the higher levels of the pyramid are not modelled in the software and are provided by people.

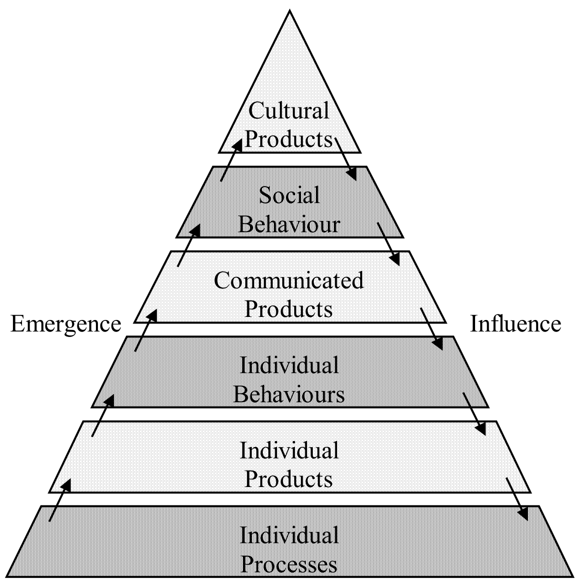


Figure 7.4: A pyramid of creativity.

Artificial creativity suggests a different approach; instead of evaluating the products of a piece of software to determine its creativity, it focuses upon the behaviours of agents and artificial societies. Artificial creativity is concerned with modelling the creative behaviours of individuals, e.g. curiosity, and studying the emergent social behaviours when individuals are put together. Because individuals in an artificial creativity simulation must be able to evaluate the creativity of communicated products and hence other individuals, the details of the products of individuals become less important. More important in the study of artificial creativity are the socio-cultural structures that emerge as a consequence of the communication of products and evaluations.

The artificial creativity approach permits the computational study of highest levels of creativity illustrated in Figure 7.4 without having to develop agents that can integrate, and achieve creative status, in human society. Artificial creativity simulations permit the experimentation with creativity in artificial societies that would be impossible in the real world, allowing the study of creativity-as-it-is in the context of creativity-as-it-could-be.

7.4 CONCLUSIONS

This chapter has presented some future directions for research using curious design agents and artificial creativity, these possibilities fall into three categories:

1) Address the limitations of the model of creative design presented here by applying curious design agents to more realistic design problems.
2) Investigate the usefulness of curious design agents in the user interface as assistants to human designers exploring complex design domains.
3) Explore the implications of developing autonomous models of creative design that can explain why their designs are creative.

This chapter has attempted to address some possible criticisms of the curious design agents presented in this thesis. It has also explored some possible applications of curious design agents in CAD systems. Some thoughts about how the model of curiosity presented in this thesis might contribute to the development of an agent that could pass a Turing Test of creativity. It is suggested that the curious exploration of design spaces provides a curious design agent with the appropriate knowledge to make a simple case for the creativity of its work.

Chapter 8

Conclusions and Future Work

This thesis started with the goal of establishing curiosity as an important new topic for design computing research. The goal was divided into subgoals within three main areas:

1) to computationally define a model of curiosity,
2) to find useful applications of the model in design,
3) to define an approach for the future study of curiosity in design.

The goals in all three of these areas have been successfully achieved.

8.1 A COMPUTATIONAL MODEL OF CURIOSITY

Chapter 4 described the computational model of curiosity used in this research and provided details of the implementations used. The computational model provides a way to augment situated agents with curiosity by monitoring the conceptual state of the agent.

Unlike other processes found in design agents, the focus of curiosity is on the conceptual state, internal to the agent, rather than the external state of the design. Curiosity is concerned with determining the interestingness of experiences from the perspective of the potential for an agent to learn. The novelty of experiences is detected by comparing the conception of a situation with expectations constructed in long-term memory from previous experiences. The interestingness of experiences can be determined from the novelty detected by applying a transformation based upon the arousal response curve found in humans when faced with novel situations, that favours similar-yet-different situations to those previously experienced.

The computational model of curiosity provides an effective way to identify those experiences that are likely to have the greatest potential for learning in a short period of time. Curiosity thus motivates design of interesting artefacts by rewarding the production of similar-yet-different experiences. The ability of curious design agents to recognise the interestingness of new experiences supports the modelling of creative design and social creativity.

8.2 APPLICATIONS OF CURIOUS DESIGN AGENTS

Several applications of curious design agents have been presented, divided by the methods used to explore design spaces and the domains studied. Curious design agents have been shown to use direct manipulation, parametric design and design tools to explore design spaces in visual and non-visual domains.

Curious design agents have been applied to the domains of rectilinear drawings, Spirograph patterns, genetic artworks and doorway design. They have shown interest in emergent shapes and patterns in visual domains and emergent crowd behaviour in pedestrian simulations.

Curious design agents have been demonstrated exploring solution space to find interesting solutions, where interesting can be either because the performance is better or worse than expected. Curious design agents have also been shown to explore problem spaces where the goal is to find problems that are interesting because they challenge the assumptions made in the design of previous solutions. In both cases, curiosity motivates the agent to explore the design space to find both “good” and “bad” examples because both help to map the design space.

8.3 ARTIFICIAL CREATIVITY

The artificial creativity approach to developing computational models of creativity in design societies, described in Chapter 2, is a useful way to study social creativity. The artificial creativity approach developed from an attempt to model Martindale’s “Law of Novelty” (Martindale, 1990), but it soon became apparent that the model could provide other insights into the behaviour of creative societies and the development of creative styles beyond Martindale’s law.

The demonstration of the artificial creativity approach in Chapter 6 shows that it provides a way to study the emergent behaviour of creative societies without the need to develop complex design agents capable of interacting with and contributing to human society. The essential element of curious design agents that make artificial creativity simulations so interesting is the ability of each of the agents to assess the potential creativity of a design and share these evaluations with others. There is no central definition of what is creative in an artificial creativity simulation and this permits the emergence of individual notions of whom and what are creative as well as the spontaneous formation of social structures such as cliques.

The artificial creativity approach opens up new avenues of research, for those interested in studying the social nature of creativity, not available in the study of human creativity and not previously considered by computational models of creative thinking. Like other fields that have adopted distributed agent simulations of social phenomena as a useful method of research (e.g. economics, computational sociology, computational anthropology), design computing stands to benefit from the adoption of the artificial creativity approach as a way of understanding creative design in its widest sense.

The ability to model creative designing in heterogeneous societies of agents, promises to expand the computational study of design to include the modelling of economic and political factors in the development of creative designs.

8.4 FUTURE RESEARCH

Artificial creativity and curious design agents provide a broad platform for future research as indicated in the previous chapter's discussion, however, several practical research directions promise considerable insights in the near future: the adaptation of existing computational models of creative design to incorporate curiosity, the construction of a distributed version of The Digital Clockwork Muse model, and the assimilation of humans within artificial creativity systems.

8.4.1 Adapting Existing Computational Design Systems

The curious design agents described in this thesis have been developed to study curious behaviour rather than the processes involved in designing. The extent to which they have achieved this goal is that they exhibit curious behaviour similar to that observed in humans during the early, conceptual, phases of design. There has been little attempt to develop full design systems capable of solving real world design problems, principally because this would have complicated the study with more complex design processes and utilitarian evaluation functions. Despite this, almost every one of the applications described in this thesis could become the subject of a much larger study.

One approach to the study of more complete applications would be to take an existing computational model of design and add an implementation of the model of curiosity presented here. The simplest way to achieve this would be to use a curious design agent to guide the evolution of an existing design tool. The evolutionary design systems that have been developed in various domains are ideal candidates. For example, Gero and Schnier's evolutionary design system (Gero and Schnier, 1995; Schnier, 1999) uses "genetic engineering" to support the evolution of new designs in specific style by constructing common building blocks found in previous examples of the style. Using a curious design agent to guide the evolutionary process would promote the evolution of "interesting", i.e. similar-yet-different, examples of the style and allow the evolution of the style beyond its original bounds in a controlled manner, complementing the learning process of both the agent and the evolutionary process.

8.4.2 Distributing the Digital Clockwork Muse

The full potential of curious agents can only be properly realised in systems comprised of multiple agents, for it is the ability of curious design agents to recognise the potential creativity of their own achievements that makes them a interesting tool for future research. The artificial creativity approach outlined in the thesis provides a method for combining many curious design agents into large artificial societies, but the implementations developed so far have been constrained by the need to run simulations within a reasonable amount of time on a single computer. An implementation goal for future research must be to break this limitation by allowing artificial creativity simulations to be run on multiple processes and across networks of computers.

The loosely coupled nature of the message passing agents developed for The Digital Clockwork Muse Project is ideal for developing distributed simulation systems and computational frameworks that would allow the development of such systems with little effort. The expectation is that massively parallel simulations of creative societies comprised of curious agents already described would permit the study of emergent social structures on a larger scale and over longer time-spans. The development of the additional simulation facilities described in the previous chapter would add even greater richness to the behaviours. In particular, the study of emergent fields and domains in large-scale systems would provide an opportunity to study the processes involved as new design disciplines emerged.

8.4.3 Assimilating Human Designers

The ultimate goal of developing distributed artificial creativity models may be to reverse the traditional role of computational design systems in human societies and allow people to be assimilated into the collective of an artificial society. Interactive versions of The Digital Clockwork Muse would provide the opportunity for experimenting with artificial societies from the inside, rather than passively observing them from the outside. The possibility of developing on-line societies of curious design agents co-operatively exploring design spaces with human designers is so different from our current notions of computational design systems that it may take a great deal of study for the true potential of these types of systems to be realised. Already there is much interest in collaborative design between people using “avatars” but artificial creativity systems presents the possibility of having computational design agents autonomously exploring design spaces and usefully interacting with human designers when they find interesting design problems or solutions.

The vision of on-line communities of computational agents interacting with human designers through a virtual reality is reminiscent of the on-line game environments that have become popular with the advent of the Internet. Similarly game-inspired research has proved successful in facilitating design communication (e.g. Maher et al., 2001). As Goertzel (1997) has noted, intelligent computers are likely to develop “design intuitions” about on-line environments that will never be achieved by human designers. Unlike humans, autonomous design agents have been designed for an existence in the virtual worlds in which they exist and therefore have been developed to interact with the world without the need to first translate their experiences into the real world in which we exist. We might reasonably expect that, in the future, truly creative design in virtual environments will accomplished by artificial societies of autonomous design agents co-operating with human clients.

8.5 SUMMARY

The research presented in this thesis has examined an array of design applications for curious design agents, from a single agent directly manipulating rectilinear sketches to artificial societies of agents collectively exploring the design space of genetic artworks. The conclusions that can be drawn are that curiosity is an important motivational force in the exploration of design spaces that can be modelled computationally to develop design agents that can autonomously explore in a familiar way. Curious design agents have the ability to explore and learn about potentially important aspects of a design space in advance of a need to apply the knowledge gained. The artificial creativity approach that has arisen from the simulation of multiple curious design agents in a social environment has suggested new directions for research of creative designing on a much larger scale than has been done before with creative design systems because it removes the need for human observers in the evaluation of designed artefacts.

Future research into the applications of curious design agents and artificial creativity is open to many directions, and, as with any complex system, the outcomes are hard to predict, but they appear to be worth pursuing, because they will undoubtedly continue to provide insights into many aspects of design and creativity and possibly provide practical systems for creative designing in both real and virtual environments.

References

Aloimonos, J., Weiss, I. and Bandyopadhyay, A. (1988) Active vision, *International Journal of Computer Vision*, **1**(4): 333–356.

Amabile, T. M. (1983) *The Social Psychology of Creativity*, Springer-Verlag, New York.

Amabile, T. M. (1985) Motivation and creativity: Effects of motivational orientation on creative writing, *Journal of Personality and Social Psychology*, **48**: 393–399.

Baker, E. (1998) *The Mug-Shot Search Problem:A Study of the Eigenface Metric, Search Strategies, and Interfaces in a System for Searching Facial Image Data*, Ph.D. Thesis, Harvard University.

Baker, E. and Seltzer, E. (1994) Evolving line drawings, *in* W. Davis and B. Joe (eds.) *Proceedings of Graphic Interface '94*, Morgan Kaufmann Publishers, pp. 91–100.

Baker, E. and Seltzer, E. (1998) The mug-shot search problem, *Proceedings of Vision Interface '98*, Canada.

Baker, D. L., Hofmann, T., McCallum, A. K. and Yang, Y. (1999) A hierarchical probabilistic model for novelty detection in text, *Technical report, Just Research*.

Baluja, S. (1996) *Expectation-Based Selective Attention*, Ph.D. Thesis, Carnegie Mellon University.

Baluja, S., Pomerleau, D. A. and Jochem, T. (1994) Towards automated artificial evolution for computer generated images, *Connection Science* **6**(2&3): 325–354.

Baraldi, A. and Alpaydin, E. (1998) *Simplified ART: A new class of ART algorithms*, International Computer Science Institute, Berkeley, CA, TR-98-004.

Bartfai, G. (1994) Hierarchical clustering with ART neural networks, *Technical Report CS-TR-94/1*, Department of Computer Science, Victoria University of Wellington, New Zealand.

Bentley, P. J. (1999a). *Evolutionary Design by Computers.* Morgan Kaufman, San Francisco, CA.

Bentley, P. J. (1999b). An introduction to evolutionary design by computers, *in* Bentley, P. J. (ed.). *Evolutionary Design by Computers*. Morgan Kaufman, San Francisco, CA, pp. 1–73.

Bentley, P.J. (1999c) From Coffee Tables to Hospitals: Generic Evolutionary Design, *in* Bentley, P. J. (ed.). *Evolutionary Design by Computers*. Morgan Kaufman, San Francisco, CA, pp. 405–423.

Bentley, P. J. (1999d). Is evolution creative? *in* P. J. Bentley and D. Corne (eds) *Proceedings of the AISB'99 Symposium on Creative Evolutionary Systems (CES)*, The Society for the Study of Artificial Intelligence and Simulation of Behaviour (AISB), pp. 28–34.

Bentley, P. J. (2000) Creative Evolution, *in* David Smith (ed.) *UCL Science 14*, University College London, pp. 14–15.

Berlyne, D. E. (1960) *Conflict, Arousal and Curiosity,* McGraw-Hill, New York.

Berlyne, D. E. (1971) *Aesthetics and Psychobiology,* Appleton-Century-Crofts, New York.

Boden. M. A. (1990) *The Creative Mind: Myths and Mechanisms.* Cardinal, London.

Boden, M. A. (1999) Computer models of creativity, *in* Robert J. Sternberg (ed.) *The Handbook of Creativity,* Cambridge University Press, UK, pp. 351–372.

Bringsjord, S. (2001) Is It Possible to Build Dramatically Compelling Interactive Digital Entertainment? *Game Studies*, **1**.

Broadbent, G. (1973) *Design in Architecture : Architecture and the Human Sciences*, John Wiley and Sons, London.

Bryant, C. H., Muggleton, S. H., Page, C. D., and Sternberg, M. J. E. (1999) Combining active learning with inductive logic programming to close the loop in machine learning, *AISB'99 Symposium on AI and Scientific Creativity*, Edinburgh, Scotland, pp. 59–64.

Cagan, J. and Agogino, A. M. (1991) Dimensional variable expansion - a formal approach to innovative design, *Research in Engineering Design* **3**: 75–85.

Carpenter, G., Grossberg, S. (1987a) A massively parallel architecture for a self-organizing neural pattern recognition machine, *Computer Vision, Graphics and Image Processing*, **37** 54–115.

Carpenter, G., Grossberg, S. (1987b) ART2: Self-organization of stable category recognition codes for analog input patterns, *Applied Optics* **26**(23) 4919–4930.

Carpenter, G., Grossberg, S. (1990) ART3: Hierarchical search using chemical transmitters in self-organizing pattern recognition architectures, *Neural Networks* **3** 129-152.

Carpenter, G., Grossberg, S., and Rosen, D. B. (1991a) ART2-A: An adaptive resonance algorithm for rapid category learning and recognition, *Neural Networks*, **4** 493–504.

Carpenter, G., Grossberg, S., and Rosen, D. B. (1991b) ARTMAP: Supervised real-time learning and classification of nonstationary data by a self-organizing neural network, *Neural Networks*, **4** 565–588.

Carpenter, G., Grossberg, S., and Rosen, D. B. (1991c) Fuzzy ART: Fast stable learning and categorization of analog patterns by an adaptive resonance system, *Neural Networks*, **4** 759–771.

Carpenter, G., Grossberg, S., Markuzon, N., Reynolds, J. H., Rosen, D. B. (1992) Fuzzy ARTMAP: A neural network architecture for incremental supervised learning of analog multidimensional maps, *IEEE Transactions on Neural Networks*, **3**(5) 698–713.

Cha, M. Y. and Gero, J. S. (1998) Shape pattern recognition using a computable shape pattern representation, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '98*, Kluwer, Dordrecht, pp. 169–188.

Clancey, W. J. (1997) *Situated Cognition: On Human Knowledge and Computer Representations,* Cambridge University Press, Cambridge, England.

Clark, A. (1994) Connectionism and cognitive flexibility, *in* T. Dartnall (ed.) *Artificial Intelligence and Creativity,* Kluwer Academic Publishers, Netherlands, pp. 63–79.

Coates, P. (1997) Using genetic programming and L-systems to explore 3D design worlds, *in* R. Junge (ed.), *CAADFutures '97*, Kluwer Academic Publishers, Netherlands.

Cohen, H. (1999) AI and Visual Creativity, *AISB '99 Convention Proceedings*, AISB, Edinburgh.

Collins, M. A. and Amabile, T. M. (1999) Motivation and creativity, *in* Robert J. Sternberg (ed.) *The Handbook of Creativity,* Cambridge University Press, UK, pp. 297–312.

Colton, S., Bundy, A. and Walsh, T. (2000a) On the notion of interestingness in automated mathematical discovery, *International Journal of Human Computer Studies,* **53**(3).

Colton, S., Bundy, A. and Walsh, T. (2000b) Agent Based Cooperative Theory Formation in Pure Mathematics, *in* G. Wiggins (ed.), *Proceedings of AISB 2000 Symposium on Creative and Cultural Aspects and Applications of AI and Cognitive Science*, Birmingham, UK, pp. 11–18.

Colton, S. and Steel, G. (1999) Artificial intelligence and scientific creativity, *Journal of the Society for Artificial Intelligence and the Study of Behaviour*, **102**: 44–50.

Crevier, D. (1993) *AI: The Tumultuous History of the Search for Artificial Intelligence*, MIT Press, Cambridge, MA.

Csikszentmihalyi, M. (1988) Society, culture, and person: a systems view of creativity, *in* R. J. Sternberg (ed.), *The Nature of Creativity,* Cambridge University Press, Cambridge, UK, pp. 325–339.

Csikszentmihalyi. M. (1996a) *Creativity: Flow and the Psychology of Discovery and Invention*, Harper Collins, New York.

Csikszentmihalyi, M. (1996b) The creative personality, *Psychology Today*, Ziff-Davis, New York, July/August: 36–40.

Csikszentmihalyi, M. (1999) Implications of a Systems Perspective for the Study of Creativity, *in* R. J. Sternberg (ed.), *Handbook of Creativity*, Cambridge University Press, Cambridge, UK, pp. 313–335.

Dacey, J. S. and Lennon, K. H. (1998) *Understanding Creativity*, Jossey-Bass, San Fransisco.

Damski, J. C. and Gero J. S., (1996) A logic-based framework for shape representation, *Computer-Aided Design,* **28**(3): 169-181.

Davis, R. and Lenat D. (1982) *Knowledge-Based Systems in AI,* McGraw-Hill Advanced Computer Science Series.

Dawkins, R. (1987) *The Blind Watchmaker*, Norton and Co., London.

Dawkins, R. (1989) The evolution of evolvability, *in* C. G. Langton (ed.) *Artificial Life*, Addison–Wesley, Reading, MA, pp. 201–220.

de Bono, E. (1986) *Lateral Thinking*, Harper and Row, New York.

Dewey, J. (1910) *How We Think*, Heath, London.

Eby, D., Averill, R., Goodman, E. and Punch, W. (1999) The optimization of flywheels using an Injection Island Genetic Algorithm, *in* P. Bentley, (ed.), *Evolutionary Design by Computers*, Morgan Kaufmann, San Francisco, pp. 167–190.

Edmonds, E. and Soufi, B. (1992) The computational modelling of emergent shapes in design, *in* Gero, J. S. and Sudweeks, F. (eds), *Computational Models of Creative Design*, The University of Sydney, pp 173–189.

Einstein, A. and Infeld, L. (1938) *The Evolution of Physics*, Simon and Schuster Press, New York.

Elman, J. (1990) Finding structure in time, *Cognitive Science*, **14**: 179-211.

Elton, M. (1995) Artificial creativity: Enculturing computers, *Leonardo*, **28**(3): 207–213

Finke, R. A., Ward, T. B. and Smith, S. M. (1992) *Creative Cognition: Theory, Research and Applications*, MIT Press, Cambridge, MA.

Franklin, S. (1997a) *Artificial Minds,* MIT Press, Cambridge, MA.

Franklin, S. (1997b) Autonomous agents as embodied AI, *Cybernetics and Systems*, **28**:6 499–520.

Franklin, S. and Graesser, A. (1996) Is it an Agent, or just a Program?: A taxonomy for autonomous agents, *in* J. P. Mueller, M. J. Wooldridge, and N. R. Jennings (eds), *Intelligent Agents III*, Springer-Verlag, Berlin, pp. 21–36.

Fritzke, B. (1996) Growing self-organizing networks – Why? *in* M. Verleysen (ed.) *ESANN'96: European Symposium on Artificial Neural Networks*, D-Facto Publishers, Brussels, pp. 61–72.

Gabora, L. (1995) Meme and variations: A computer model of cultural evolution, *in* L. Nadel & D. Stein, (eds) *1993 Lectures in Complex Systems*, Addison-Wesley, Reading.

Gabora, L. (1996) Culture, evolution, and computation, *in* T. Furuhashi, (ed.) *Proceedings of the Second Online Workshop on Evolutionary Computation,* Society of Fuzzy Theory and Systems, pp. 9–12.

Gabora, L. (1997) The origin and evolution of culture and creativity. *Journal of Memetics: Evolutionary Models of Information Transmission* **1**(1): 1–28.

Gabora, L. (2000) The beer can theory of creativity, *in* P. Bentley & D. Corne, (eds) *Creative Evolutionary Systems*, Morgan Kauffman, San Francisco, CA, pp. 1–26.

Gardner, H. (1993) *Creating Minds,* Basic Books, New York.

Gaver, W. W. and Mandler, G. (1987) Play it again, Sam: On liking music, *Cognition and Emotion*, **1**(3): 259–282.

Gero, J. S. (1994b) Computational models of creative design processes, *in* T. Dartnall (ed.) *Artificial Intelligence and Creativity,* Kluwer Academic Publishers, Netherlands, pp. 269–281.

Gero, J. S. and Jun, H. J., (1995) Visual semantics emergence to support creative designing: a computational view, *in* Gero, J. S., Maher, M. L. and Sudweeks, F. (eds), *Preprints Computational Models of Creative Design*, Key Centre of Design Computing, Department of Architectural and Design Science, The University of Sydney, Sydney, pp. 87–116

Gero, J. S. and Saunders, R. (2000) Constructed representations and their functions in computational models of designing, *in* B–K. Tang, M. Tan and Y–C. Wong (eds), *Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2000),* CASA, Singapore, pp. 215–224.

Gero, J. S. and Schnier, T. (1995). Evolving representations of design cases and their use in creative design, *in* J. S. Gero, M. L. Maher and F. Sudweeks (eds), *Preprints Computational Models of Creative Design*, Key Centre of Design Computing, University of Sydney, pp. 343–368.

Gero, J. S. and Yan, M., (1993) Discovering emergent shapes using a data-driven symbolic model, *in* Flemming, U. and Van Wyk, S. (eds), *CAAD Futures '93*, North Holland, Amsterdam, pp. 3–17.

Goertzel, B. (1997) *From Complexity to Creativity*, Plenum Press, New York.

Goldberg, D. E. (1999) The race, the hurdle and the sweet spot: Lessons from genetic algorithms for the automation of design innovation and creativity, *in* Bentley, P. J. (ed.) *Evolutionary Design by Computers*, Morgan Kaufman, San Francisco, CA, pp. 105–118.

Gomes, P., Bento C., and Gago O. (1999) A function for novelty evaluation in design, *in* J. S. Gero and M. L. Maher (eds) *Computational Models of Creative Design IV,* Key Centre of Design Computing and Cognition, University of Sydney, pp. 19–31.

Graf, J. and Banzhaf, W. (1995) Interactive evolution of images, *in* J. R. McDonnell, R. G. Reynolds, and D. B. Fogel (eds) *Evolutionary Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*.

Grossberg, S. (1976) Adaptive pattern classication and universal recoding, II: Feedback, expectation, olfaction and illusions, *Biological Cybernetics* **23**: 187–202.

Grossberg, S. and Mingolla, E. (1985a) Neural dynamics of form perception: Boundary completion, illusory figures, and neon color spreading, *Psychological Review,* **92**: 173–211.

Grossberg, S. and Mingolla, E. (1985b) Neural dynamics of perceptual grouping: Textures, boundaries, and emergent segmentation, *Perception & Psychophysics*, **38**: 141–171.

Guilford, J. P. (1967) *The Nature of Human Intelligence*, McGraw-Hill, New York.

Haag, G. and Liedl, P. (2001) Modelling and simulating innovation behaviour within micro-based correlated decision processes, *Journal of Artificial Societies and Social Simulation*, **4**(3), http://www.soc.surrey.ac.uk/JASSS/4/3/3.html.

Hayes-Roth, F. (1983) Using proofs and refutations to learn from experience, *Machine Learning: An Artificial Intelligence Approach*, Springer-Verlag, Berlin, pp. 221–240.

Helbing, D. (1991) A mathematical model for the behavior of pedestrians *Behavioral Science*, **36**: 298–310.

Helbing, D. and Molnár, P. (1995) Social force model for pedestrian dynamics, *Physical Review E*, **51**: 4282–4286.

Helbing, D. and Molnár, P. (1997) Self-organization phenomena in pedestrian crowds, *in* F. Schweitzer (ed.) *Self-Organization of Complex Structures: From Individual to Collective Dynamics*, Gordon and Breach, London, pp. 569–577.

Hofstadter, D. (1979) *Godel, Escher, Bach: An Eternal Golden Braid*, Basic Books, New York.

Hofstadter, D. and the Fluid Analogies Research Group (1995a) *Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, Basic Books, New York.

Hofstadter, D. (1995b) Creativity, brain mechanisms, and the Turing Test, *in* D. Hofstadter (ed.) *Fluid Concepts & Creative Analogies: Computer Models of the Fundamental Mechanisms of Thought*, Basic Books, New York, pp. 467–491.

Home, H. (1765) *Elements of Criticism*, Millar, London.

Humphrey, N. K. (1973) The illusion of beauty, *Perception*, **2**: 429–439.

Kaski, S., Kangas, J. and Kohonen, T. (1998) Bibliography of self-organizing map (SOM) papers: 1981-1997, *Neural Computing Surveys*, **1**(3&4): 1–176.

Koestler, A. (1964) *The Act of Creation,* Macmillan, New York.

Kohonen, T. (1993) *Self-Organization and Associative Memory*, 3rd ed., Springer, Berlin.

Kohonen, T. (1995) *Self-Organizing Maps*, Springer–Verlag, Berlin.

Koning, H. and Eizenberg, J. (1981) The language of the prairies: Frank Lloyd Wright's Prairie Houses, *Environment and Planning B*, **8**: 295–323.

Koza, J. R. (1992) *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press, Cambridge, MA.

Koza, J. R., Bennett III, F. H., Andre, D., and Keane, M. A. (1999) *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann, San Francisco, CA.

Langley, P., Simon, H. A., Bradshaw, G. L., and Zytkow, J. M. (1987) *Scientific Discovery: Computational Explorations of the Creative Processes,* MIT Press, Cambridge, MA.

Langton, C. G. (1989) Artificial life, *Artificial Life*, Addison–Wesley, Reading, pp. 1–47.

Lenat, D. B. (1976) *AM: An artificial intelligence approach to discovery in mathematics,* Ph.D. thesis, Stanford University.

Lenat, D. B. (1983) EURISKO: A program that learns new heuristics and domain concepts, *Artificial Intelligence,* **21**:61–98.

Lenat, D. B. and Brown, J. S. (1984) Why AM and EURISKO appear to work, *Artificial Intelligence,* **23**:269–294.

Lewis, M. (2000) Aesthetic evolutionary design with data flow networks, *Generative Art 2000*, Milan.

Lipson, H. and Pollack, J. B. (2000) Automatic design and manufacture of artificial lifeforms, *Nature*, **406**: 974–978.

Lindsay, R., Buchanan, B., Feigenbaum, E., and Lederberg, J. (1993) DENDRAL: A case study of the first expert system for scientific hypothesis formation, *Artificial Intelligence*, **61**(2): 209–261.

Liu, Y–T., (1993) A connectionist approach to shape recognition and transformation, *in* Flemming, U. and Van Wyk, S. (eds), *CAAD Futures '93*, North Holland, Amsterdam, pp. 19–36.

Liu, Y. T. (2000) Creativity or novelty? *Design Studies* **21**(3): 261–276.

Macedo, L. and Cardosa, A. (2001a) Creativity and surprise, *in* G. Wiggins (ed.), *AISB'01 Symposium on AI and Creativity in Arts and Science*, SSAISB, York, UK.

Macedo, L. and Cardosa, A. (2001b) SC-EUENE – Surprise/curiosity-based exploration of uncertain and unknown environments, *AISB'01 Symposium on Emotion*, SSAISB, York, UK.

Maes, P. (1994) Agents that reduce work and information overload, *Communications of the ACM*, **37**(7): 31–40.

Maher, M. L., Gu, N. and Li, F. (2001),Visualisation and object design in virtual architecture, *in* J. S. Gero, S. Chase and M. Rosenman (eds), *CAADRIA2001*, Key Centre of Design Computing and Cognition, University of Sydney, pp. 39–50.

Major, M. D., Stonor, T. and Penn, A. (1998) Passengers, pedestrians and shoppers: Space syntax in design, *Passenger Terminal World*, April.

Mandelbrot, B. B. (1977) *The Fractal Geometry of Nature*, W. H. Freeman & Co., New York.

Marsland, S., Nehmzow, U. and Shapiro, J. (2000a) Novelty detection for robot neotaxis, *International Symposium On Neural Computation (NC'2000)*, pp. 554–559.

Marsland, S., Nehmzow, U. and Shapiro, J. (2000b) A real–time novelty detector for a mobile robot, *EUREL European Advanced Robotics Systems Conference – Robotics 2000,* Salford, England.

Marsland, S., Nehmzow, U. and Shapiro, J. (2000c) Novelty detection on a mobile robot using habituation, *From Animals to Animats, The Sixth International Conference on Simulation of Adaptive Behaviour*, Paris, France.

Marsland, S., Nehmzow, U. and Shapiro, J. (2001) Novelty detection in large environments, *Proceedings of Towards Intelligent Mobile Robots 2001 (TIMR'01)*, Bristol, England.

Martindale, C., Moore, K., and West, A. (1988) Relationship of preference judgements to typicality, novelty, and mere exposure, *Empirical Studies of the Arts,* **6**(1).

Martindale, C. (1990) *The Clockwork Muse*, Basic Books, New York.

Newell, A., Shaw, J. C. and Simon, H. A. (1962) The process of creative thinking, *in* H. Gruber, G. Terrell and M. Wertheimer (eds), *Contemporary Approaches to Creative Thinking,* Atherton Press, New York, pp. 63–119.

Nwana, H. (1996) Software agents: An overview. *The Knowledge Engineering Review* **11**(3).

Partridge, D. Rowe, J. (1994) *Computers and Creativity*, Intellect Books, Oxford.

Peters, M. (2000) *Making Sense: Autonomy and adaptation in visual robotics*, Ph.D. Thesis, The University of New South Wales, Australia.

Pollack, J. B. (1990) Recursive distributed representations, *Artificial Intelligence* **46**(1): 77–105.

Purcell, T. A. and Gero, J. S. (1998) Drawings and the design process, *Design Studies* **19**: 389–430.

Reffat, R. and J. S. Gero (1998) Learning about shape semantics: A situated learning approach, *in* T. Sasada, S. Yamaguchi, M. Morozumi, A. Kaga and R. Homma (eds), *CAADRIA'98*, CAADRIA, Kumomoto, Japan, pp. 375–384.

Ritchie, G. D. and Hanna, F. K. (1984) AM: A case study in AI methodology, *Artificial Intelligence Journal*, **23**: 249–263.

Rivest, R. L., Shamir, A., and Adleman, L. (1978) A method for obtaining digital signatures and public key cryptosystems, *Communications of the ACM* **21**(2): 120–126.

Rosenman, M. A. (1996) A growth model for form generation using a hierarchical evolutionary approach, *Microcomputers in Civil Engineering*, special issue on Evolutionary Systems in Design, **11**(3): 161–172.

Rosenman, M. A. and Gero, J. S. (1993) Creativity in design using a design prototype approach, *in* J. S. Gero and M. L. Maher (eds), *Modeling Creativity and Knowledge-based Creative Design*, Lawrence Erlbaum, Hillsdale, New Jersey, pp.111–138.

Rowe, J. and Partridge, D. (1993) Creativity: A survey of AI approaches, *Artificial Intelligence Review*, **7**: 43–70.

Schmidhuber, J. (1991a) Adaptive confidence and adaptive curiosity, *Technical Report FKI-14991*, Technische Universitat München, München, Germany.

Schmidhuber, J. (1991b) A possibility for implementing curiosity and boredom in model-building neural controllers, *in* J. A. Meyer and S. W. Wilson (eds) *Proceedings of the International Conference on Simulation of Adaptive Behaviour: From Animals to Animats,* MIT Press, Cambridge, MA. pp. 222–227.

Schmidhuber, J. (1991c) Curious model-building control systems, *in Proceedings of the International Joint Conference on Neural Networks*, Singapore, IEEE, volume 2, pp. 1458–1463.

Schmidhuber, J. (1997) What's interesting? *Technical Report TR-35-97*, IDSIA, Lugano, Switzerland.

Schnier, T. and Gero, J. S. (1996) Learning genetic representations as alternative to hand-coded shape grammars, *in* J. S. Gero and F. Sudweeks (eds), *Artificial Intelligence in Design '96*, Kluwer, Dordrecht, pp. 39–57.

Schnier, T. (1999) *Evolved Representations and Their Use in Computational Creativity*, Ph.D. Thesis, The Key Centre of Design Computing, University of Sydney.

Schön, D. A. (1983) *The Reflective Practitioner: How Professionals Think in Action,* Basic Books, New York.

Schön, D. A. and Wiggins, G. (1992) Kinds of seeing and their functions in designing, *Design Studies,* **13**: 135–156.

Schulman, E. R. and Cox, C. V. (1997) How to write a Ph.D. dissertation, *Annals of Improbable Research*, **3**(5): 8.

Scott, P. D., and Markovitch, S. (1989a) Learning Novel Domains Through Curiosity and Conjecture, *Proceedings of International Joint Conference for Artificial Intelligence*, Detroit, Michigan, pp. 669–674.

Scott, P. D., and Markovitch, S. (1989b) Uncertainty Based Selection of Learning Experiences, *in Proceedings of The Sixth International Workshop on Machine Learning*, Ithaca, New York, pp. 358–361.

Silberschatz, A. and Tuzhilin, A. (1996) What makes patterns interesting in knowledge discovery systems, *IEEE Transactions on Knowledge and Data Engineering* **8**(6): 970–974.

Sims, K. (1991) Artificial evolution for computer graphics. *Computer Graphics* **25**(4): 319-328.

Stanley, J. C. (1976) Computer simulation of a model of habituation. *Nature,* **261**: 146–148.

Sternberg, R. J. (1988) *The Nature of Creativity,* Cambridge University Press, Cambridge, UK.

Sternberg, R. J. and Lubart, T. I. (1999) The Concept of Creativity: Prospects and Paradigms, *in* Robert J. Sternberg (ed.), *The Handbook of Creativity,* Cambridge University Press, Cambridge, UK, pp. 3–15.

Suwa, M., Gero, J. S. and Purcell, T. (1999) Unexpected discoveries and S-invention of design requirements: A key to creative designs, *in* J. S. Gero and M. L. Maher (eds), *Computational Models of Creative Design IV, Key Centre of Design Computing and Cognition, University of Sydney,* pp. 297–320.

Tabuada, P., Alves, P., and Gomes, J. (1998a) Artificial art through genetic algorithms, *Proceedings of Genetic Algorithms and Artificial Life '98 (GAAL'98)*, pp. 46–49.

Tabuada, P., Alves P., Gomes, J. and Rosa, A. (1998b) 3D artificial art by genetic algorithms, *Proceedings of the Workshop on Evolutionary Design at Artificial Intelligence in Design '98 (AID'98)*, pp. 18–21.

Taylor, C. W. (1988) Various approaches to the definition of creativity, *in* R. J. Sternberg (ed.), *The Nature of Creativity,* Cambridge University Press, Cambridge, UK, pp. 99–124.

Taylor, R. P., Micolich, A. P. and Jonas, D. (1999) Fractal analysis of Pollock's drip paintings, *Nature*, **399**:422.

Thaler, S. (1996) Neural networks that autonomously create and discover, *PC AI*, May/June.

Thompson, A. (2000) Explorations in design space: Unconventional electronics design through artificial evolution, *IEEE Transactions on Evolutionary Computing*, **3**: 167–196.

Todd, S. and Latham, W. (1992) *Evolutionary Art and Computers*. Academic Press, London.

Tomlinson, P. and Gero, J. S. (1997), Emergent shape generation via the boundary contour system, *in* R. Junge (ed.), *CAAD Futures '97*, Kluwer, Dordrecht, pp. 865–874.

Turing, A. M. (1950) Computer machinery and intelligence, *Mind*, **LIX**(236): 433–460.

Ullman, S. (1984) Visual routines, *Cognition*, **18**: 97–159.

Waltz, D. (1998) The Importance of importance, *AI Magazine*, **20**(3): 18–35

Wallas, G. (1926) *The Art of Thought,* Harcourt, Brace, Jovanovich, New York.

Weenink, D. (1997) Category ART: A variation on Adaptive Resonance Theory neural networks, *Proceedings of the Institute of Phonetic Sciences 21*, pp. 117–129.

Whitfield, A. and Wiltshire, J. (1982) Design training and aesthetic evaluation: An intergroup comparison, *Journal of Environmental Psychology*, **2**: 109–117.

Witbrock, M. and Reilly, S. N. (1999) Evolving genetic art, *in* Bentley, P. J. (ed.) *Evolutionary Design by Computers*, Morgan Kaufman, San Francisco, CA, pp. 251–259.

Wooldridge, M. J. and Jennings, N. R. (1995) Intelligent agents: Theory and practice, *Knowledge Engineering Review,* **10**(2): 115–152.

Wooldridge, M. J. (1999) Intelligent agents, *in* G. Weiss (ed.), *Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*, MIT Press, Cambridge, MA, pp. 27–77.

Ziemke, T. (2000) *Situated Neuro-Robotics and Interactive Cognition*, Doctoral Dissertation, Department of Computer Science, University of Sheffield, UK

Appendix A

Publications Arising from this Research

Constructed Representations and their Functions in Computational Models of Designing

Gero, J. S. and Saunders, R. (2000) Constructed representations and their functions in computational models of designing, *in* B–K. Tang, M. Tan and Y–C. Wong (eds), *Proceedings of the Fifth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2000),* CASA, Singapore, pp. 215–224.

This paper re-examines the conclusions made my Schön and Wiggins in 1992 that computers were unable to reproduce processes crucial to designing. We propose that recent developments in artificial intelligence and design computing put us in a position where we can begin to computationally model designing as conceived by Schön and Wiggins. We present a computational model of designing using situated processes that construct representations. We show how constructed representations support computational processes that model the different kinds of seeing reported in designing. We also present recently developed computational processes that can identify unexpected consequences of design actions using adaptive novelty detection.

Designing for Interest and Novelty: Motivating Design Agents

Saunders, R. and Gero, J. S. (2001a) Designing for interest and novelty: motivating design agents, *in Proceedings of CAAD Futures 2001*, Eindhoven.

This paper is concerned with the motivation of design agents to promote the exploration of design spaces. A general form of motivation common to designers is a curiosity to discover interesting designs. This paper presents computational models of interest and curiosity based on the detection of novelty. We illustrate the behaviour of our model of interest by developing a design agent that is motivated to explore the effects of emergent crowd behaviours on the performance of doorways.

A Curious Design Agent: A Computational Model of Novelty-Seeking Behaviour in Design

Saunders, R. and Gero, J. S. (2001b) A curious design agent: A computational model of novelty-Seeking behaviour in design, *in Proceedings of the Sixth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2001)*, The University of Sydney, Australia.

This paper presents a "curious design agent", i.e. an agent that uses the search for novel designs to guide its design actions. A computational model of curiosity based on a process called novelty detection is presented. The behaviour of the computational model is illustrated with a curious design agent searching the space of two-dimensional patterns generated by a simulated Spirograph is reported.

The Digital Clockwork Muse: A Computational Model of Aesthetic Evolution

Saunders, R. and Gero, J. S. (2001c) The Digital Clockwork Muse: A computational model of aesthetic evolution, *in* G. Wiggins (ed.), *Proceedings of the AISB'01 Symposium on AI and Creativity in Arts and Science*, SSAISB, York, UK.

This paper presents a computational model of creativity that attempts to capture within a social context an important aspect of the art and design process: the search for novelty. The computational model consists of multiple novelty-seeking agents that can assess the interestingness of artworks. The agents can communicate to particularly interesting artworks to others. Agents can also communicate to reward other agents for finding interesting artworks. We present the results from running experiments to investigate the effects of searching for different degrees of novelty on the artworks produced and the social organisation of the agents.

Artificial Creativity: A Synthetic Approach to the Study of Creative Behaviour

Saunders, R. and Gero, J. S. (to appear) Artificial creativity: A synthetic approach to the study of creative behaviour, *Fifth International Roundtable Conference on Computational and Cognitive Models of Creative Design, Heron Island.*

We present a novel approach to the computational study of creativity, called Artificial Creativity. Artificial Creativity promotes the study of the creative behaviour of individuals and societies in artificial societies of agents. It is similar to the approach to that taken by Artificial Life researchers involved in developing computational models. We present a framework for developing Artificial Creativity systems as an adaptation of Liu's dual generate-and-test model of creativity. An example implementation of an Artificial Creativity system is presented to illustrate the potential benefits of our new approach as a way of investigating the emergent nature of creativity in societies of communicating agents. Finally, we discuss some future research directions that are possible by extending the abilities of individuals and studying the emergent behaviour of societies.

Appendix B

Instructions for Using the Enclosed CD-ROM

The CD-ROM enclosed with this thesis has some example applications and videos to illustrate the domains explored by curious design agents. The CD-ROM also contains a complete copy of the thesis in Adobe Acrobat format, i.e. PDF. To view this material the Acrobat Reader browser plug-in is required. The plug-in can be downloaded from Adobe's website: http://www.adobe.com/

The CD-ROM is organised as a small website; the easiest way to explore the contents is using a modern web browser. To view the applets the browser must support Java v1.1 or later, e.g. Microsoft Internet Explorer 5.0+. To explore the contents of the CD-ROM open the file `index.html` in the root directory. This file contains the main menu for the CD-ROM and allows access to the various documents, applets, and videos.

8.6 CONTENTS

The main menu is divided into three sections: Thesis, Applets and Videos. The first section contains the full text to this thesis in Adobe Acrobat format. The thesis is available either as a single file containing the whole document or as separate chapters. If the Acrobat Reader plug-in has been installed correctly then the files should open automatically when they are clicked on. If there is any problem accessing these files through a web browser, they can be located in the directory called "thesis" for viewing outside of a browser.

The applets have been divided into three types. First, there are demonstrations of neural network learning using a simple 2D input space as described in Chapter 4. Second, there are a couple of applets that allow the reader to explore some of the domains explored by curious design agents in Chapter 5 & 6. Finally, there are two examples of curious agents; the first curious agent was discussed in Chapter 7, the second has not been presented in this thesis but may be interesting for readers that wish to observe curious behaviour in a very simple environment.

The video section contains a link to a selection of videos that illustrate some typical examples of the emergent crowd behaviour explored by the curious design agent in Section 6.1. The videos have been provided in Microsoft AVI format and Apple QuickTime. To view these videos you will require either Windows Media Player or Apple's QuickTime movie player.