Towards a Computational Model of Creative Cultures

Rob Saunders and **Kazjon Grace**
University of Sydney, Australia
`[rob|kgra7942]@arch.usyd.edu.au`

Abstract

This paper examines the possibility for developing a computational model of creative systems that supports the evolution of language in creative cultures. First we present some previous work developing models of creative individuals and creative societies. The extension of these earlier models towards a model of creative cultures is discussed with particular emphasis on the modelling of the evolution of language in creative domains using "language games." The model presented here extends the previous work with the potential for incorporating aspects of the cultural situation that affect the production, evaluation and adoption of creative works. We conclude with a discussion of the potential significance of developing language capable multi-agent systems for the modelling of creative cultures.

Creative behaviour is personally, socially and culturally situated: creative individuals work within environments rich with personal experiences, social relationships and accumulated cultural knowledge. The majority of computational models of creativity have focussed on the cognitive and individual levels of the creative process, some work has begun to model social factors but little work has attempted to model creativity within a cultural environment. Developing models that incorporate a cultural component will allow the affects of culture on creative development to be studied in simulation—potentially creating new tools for the study and support of creativity.

Traditionally, computational models of creativity have focussed on capturing individualistic, or cognitive, aspects by modelling the creative process as generative systems, e.g., models of anology-making (French 2006). Often these have been developed based on Newell & Simon's characterization of creativity as problem-solving (Newell, Shaw, & Simon 1958; Newell & Simon 1972), however, as some commentators have pointed out, creative individuals can be distinguished as much by their ability to find new problems as by their capacity to solve problems posed by others (Getzels & Csikszentmihalyi 1976).

Studies of human creativity has revealed the importance of the social and cultural environment for creative individuals (Gruber 1981; Martindale 1990; Gardner 1993). Creative individuals are often drawn to particular locations at the intersections of cultures because they offer the richest opportunities for novel and diverse cultural experiences. For example, Vienna at the beginning of the 20th Century was a fertile environment for creative ideas because of its geographic location at the crossroads of Eastern and Western cultures. Developing computational models that support distinct cultural environments, e.g., separate languages, will allow us to explore a variety of questions about how these novel experiences may arise *in silico*.

Csikszentmihalyi's systems view of creativity, later developed into the Domain Individual Field Interaction (DIFI) framework presented in Feldman, Csikszentmihalyi, & Gardner (1994), is a unified approach to studying human creativity that provides an integrated view of individual creativity within a social and cultural context (Csikszentmihalyi 1988). According to this framework, a *creative system* has three interactive subsystems: *domain*, *individual* and *field*. A domain is an organised body of knowledge, including specialised languages, rules, and technologies. An individual is the generator of new works in a creative system, based on their knowledge of the domain. A field contains all individuals who can affect the content of a domain, e.g., creators, audiences, critics, educators, etc.

The interactions between individuals, fields and domains (illustrated in Figure 1) form the basis of the creative process in the DIFI framework: individuals acquire knowledge from domains and propose new knowledge evaluated by the field; if the field accepts a proposed addition, it becomes part of the domain and available for use by other individuals.

As a first step towards developing a model of culturally situated creativity, we are focussing on developing a model of the evolution of specialised languages associated with one or more creative domains. In doing so, we are exploring the ways in which the evolution of language and creative behaviour of individuals and fields affect each other. How does creative behaviour affect the evolution of language? How does the evolution of language affect creative behaviour?

Previous Work

Previous computational models of creative individuals attempted to model some of the motivations that drive individuals to be creative (Saunders & Gero 2001). Curious agents embody a model of curiosity based on studies of humans and

---

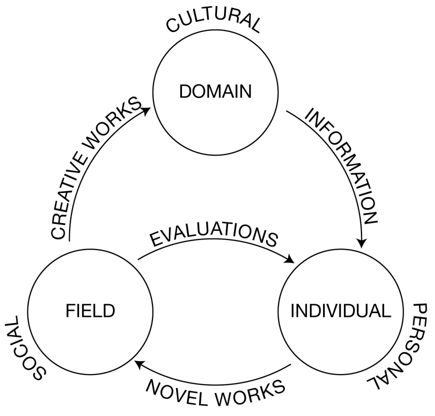


Figure 1: The DIFI model of creative systems

other animals, where curiosity is triggered by a perceived lack of knowledge about a situation and motivates behaviour to reduce uncertainty through exploration (Berlyne 1971).

Unlike other models of creative processes that try to maximise some utility function, curious agents are motivated to discover 'interesting' works based on their previous experiences. Interest is calculated according to an "hedonic function" based on novelty where the most interesting works are similar-but-different to those that have been experienced before, see Figure 2.

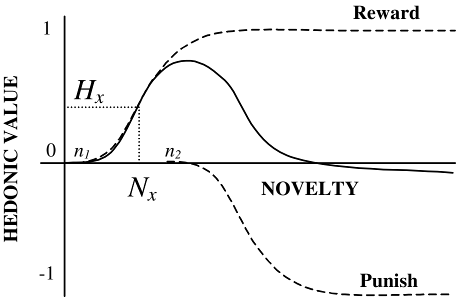


Figure 2: The Wundt Curve, an hedonic function used to model curiosity in curious design agents.

Using the DIFI framework as a guide, curious agents have been used to develop computational models of creative fields to investigate emergent social phenomena, e.g., the formation of cliques (Saunders & Gero 2002). Using curious agents to model creative fields produced a simple model where curious agents share 'interesting' works with peers for evaluation. Interestingness is judged against an agent's memory of previously seen works, including those that it has produced and those sent to it by other agents.

In this model, pairs of agents engage in a simple interaction where an initiator, having already produced a work that it believes to be 'interesting', sends it to a second agent for evaluation. The second agent evaluates the work received against its memory of previously seen works and responds with an evaluation of the work as a numerical value of how 'interesting' the work is.

Works that are determined to be 'interesting' by individuals other than the creator, i.e., by agents asked to evaluate a work, can be added to a central repository of 'creative' works. By not allowing creators to directly add their works to the repository, some level of agreement about what constitutes an 'interesting' work must be achieved. This repository provided a store of works within the domain, but this is only one small part of the domain's role in a creative system.

This computational model of a creative system supported the simulation of some general patterns observed in human creativity, e.g., the isolation of individuals that fail to innovate at an appropriate rate to gain acceptance by a field (Martindale 1990). An example of a social simulation is shown in Figure 3 where two individuals have been ignored by a field, one for innovating too slowly, the other for innovating too quickly (Saunders & Gero 2002).

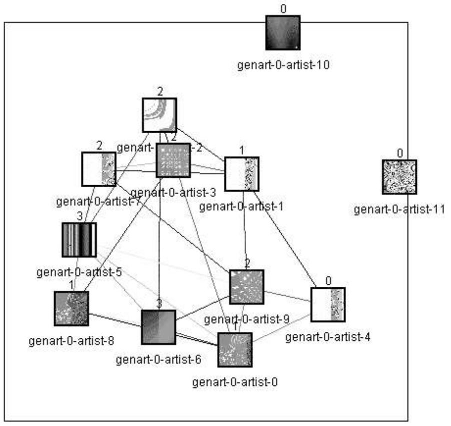


Figure 3: The Digital Clockwork Muse, a computational model of social creativity using curious design agents.

These early models of creative systems incorporated a simple model of the domain, where highly-valued works could be added to a repository if found sufficiently interesting by agents other than the creator. This model of the domain as a store of examples of valuable works that can be used by individuals as starting points for the production of new works. This model of the domain is insufficient to model most cultural aspects of interest, mostly because of the lack of a domain specific language.

The research project presented here extends previous work by incorporating additional aspects of the cultural situation that affect the production, evaluation and adoption of creative works. Significantly, this involves the development of a new model that incorporates the evolution of domain-specific languages in creative cultures.

In contrast to existing social models, such a cultural model will allow agents to record traces of their interactions, experiences and related artefacts. These records and descriptions serve as starting points for agents to study and develop new knowledge as well as to transfer knowledge between domains. The proposed model represents the next step towards modelling the cycle of production in creative systems; from the domain to the individual to the field and back to the domain. For example, the use of language allows individuals to describe new creative ideas in advance of producing an artefact, thereby setting a goal for the production of the artefact. Our aim is to develop agents that can use language to pose 'interesting' creative challenges as ambiguous linguistic constructions.

The Proposed Model

In our proposed model of a creative system, agents continue to share works with peers in a field as before, sending interesting works for evaluation to other agents. In addition, agents can communicate descriptions of works, e.g., requirements, as linguistic expressions. This extends the previous model of social creativity and places more emphasis on the importance of interactions in creative systems, in line with the DIFI framework.

To develop our new model, we have adapted the peer-to-peer interaction between individuals to incorporate a model of the evolution of language in creative cultures on the work of Steels, who developed a model of the evolution of language as a consequence of a population of agents playing 'language games' between pairs of agents (Steels 1996b). Such models are capable of producing lexicons of words with meanings grounded in experience.

The evolution of language is a distributed and self-organizing; through the repeated playing of language games between pairs of agents, a shared lexicon of words and their associated meanings evolve in combination. Of particular interest, from our perspective of modelling domain-specific languages, are the ambiguities that arise in the languages evolved by playing language games; a single word may have multiple meanings and multiple words may have the same meaning.

Anyone who has tried to communicate across disciplinary boundaries will likely have experienced something similar, e.g., familiar words having unfamiliar meanings and unfamiliar words having familiar meanings. The resolution of tensions created when individuals from different fields communicate has the potential for creative output as the meanings of words are negotiated (Gemeinboeck & Dong 2006).

In our proposed model, we extend the existing models of the evolution of language by developing a new language game based on the interactions within a creative systems, where *speaker* and *listener* become *creator*, *audience* and *critic*.

Language Games

A language game is an abstract and simplified method of communication. The concept was introduced by Ludwig Wittgenstein to study the use of language in society (Wittgenstein 1953). A simple example of a language game, coined by Wittgenstein, is a dialogue between a builder and his assistants. The builder identifies what type of stone is needed for construction, by saying "slab", "block" or "beam", and the assistant responds by bringing the appropriate materials in the appropriate order. Wittgenstein describes language games in which participants can communicate to describe or learn about objects, report events, give commands or solve problems.

As a tool in multi-agent system development, language games enable the dissemination of concepts and biases throughout a population if language games in which pressure exists to develop a coherent lexicon are played. In a society of curious design agents, we are using language games to model the development of specialised languages within fields as the agents explore the domain for 'interesting' works.

Systems based on the evolution of language based on language games is that the system is open, agents can be added or removed from the system at any time. Agents that are added to a system can quickly adapt to the lexicon in use. We plan to use this capacity to develop models of multiple domains that agents can move between. This type of movement will allow agents to both adapt to the lexicons used in different domains but also affect the development of language as it transports meanings and words from one domain to another.

Guessing Games

The language games of Steels are guessing games, where one agent describes an object to another and the second attempts to identify the correct topic from the description (Steels 1995). Steels guessing games have been used to discriminate between different agents (Steels 1996b), between other objects in the agents context (Steels 1996a) and between real objects as presented to a pair of robotic "Talking Heads" (Steels 1998). In the course of attempting to succeed at as many language games as possible, the society of agents is driven to adopt common meanings for their initially random words and a shared lexicon emerges. Steels uses this model to support the position that language is an autonomous adaptive system and that its emergence in humans could have been the result of self-organisation rather than the acquisition of a specific language-capable area of the brain.

Imitation Games

Another form of language game, an imitation game, has been used to explore the self-organization of vowel systems (de Boer 2000) and the evolution of music (Miranda, Kirby, & Todd 2003). These systems share with the research presented in this paper the quality of having a production component that is guided by language games. In the case of the adaptive vowel systems the agents imitate each other by producing an expression of the sounds they perceive. In the society of musical agents, compositions are shared through agents performing for each other.

The goal of the vowel-formation system (de Boer 2000) was to investigate the structural tendencies of vowel systems and to determine where a coherent vowel system could emerge in a simulation. The agents were given articulatory and acoustic sensors modelled on human speech and hearing capabilities. An agent would choose a set of vowels at random and utter them to another agent that would interpret them according to its vowel system and produce an imitation. The initiator would then indicate whether the imitation (as understood by the initiator) was the same as the original signal. The agents were found to be capable of evolving coherent vowel systems, and it was found that this ability was robust to background noise.

In the model of the evolution of music, a similar mechanism to that used by de Boer is used to promote the propagation of, initially randomly generated, tunes throughout a society (Miranda, Kirby, & Todd 2003). The success of a tune is, like the success of a vowel, measured by the ability of another agent to reproduce it. A tune is successfully reproduced when the agent who produced the initial performance knows no tunes that are more similar to the imitators recital than the one it initially performed. The agents quickly developed coherent sets of tunes and were capable of successful recitals.

Generation Games

To allow us to model a common form of interaction in design, we are introducing a new type of language game, which we call a *generation game*. A generation game involves a request by a "client" (initiator) agent to a "designer" agent. The request encodes a set of design requirements specified by the client, the job of the design agent is to generate a satisfactory design, i.e., a design that satisfies the requirements. Unlike the vowel and music imitation games, where the requirements and the expression are the same thing, in the generation game the expressed requirements contain a set of features that a design must contain but the specific form of the design will depend on the generative capabilities of the design agent. To succeed, the design agent must generate a design that satisfies the requested feature set by exploring the space of possible designs for a design that satisfies the design requirements as it understands them.

Unlike the guessing game and the imitation game, there may be many possible designs that satisfy a single design requirement. This opens the possibility for judging success or failure on more than just the ability of a design to satisfy a set of required features, but to have an implicit requirement for all designs to be 'interesting', according to some function of interest that does not contradict the intended meaning of words within a lexicon. Consequently, it is our intention to show that it is possible, without undermining the grounding of words within an evolving language, to integrate language games and curious design agents.

Formal Description

This section provides formal descriptions of a guessing game, adapted from Steels (1996b; 1996a), and the proposed generation game for modeling client-designer interactions. In the following sections we will discuss how both of these language games may be used to model the evolution of language in creative domains.

The Guessing Game

In the guessing game, agents attempt to identify the object being described by another agent from a context of objects. Let there be a set of objects $O = (o_1, ..., o_m)$ and a set of sensory channels $\Sigma = (\sigma_1, ..., \sigma_n)$ which are real valued functions over O. Each function σ_j defines a value for each object o_i. Each agent a has a set of feature detectors, or sensors, $S_a = (s_{a,1}, ..., s_{a,m})$. A sensor $s_{a,k}$ consists of a set of possible values $V_{a,k}$, a function $\phi_{a,k}$ and a sensory channel σ_j. The result of applying a feature detector $s_{a,k}$ to an object o_i is a value $v = \phi_{a,k}(\sigma_j(o_i)) \in V_{a,k}$.

The feature set derived by applying the feature detectors of an agent a to an object o_i is defined as F_{a,o_i}. A distinctive feature set D^C_{a,o_i} is a set of features that serve to uniquely identify an object o_i to agent a from a set of other objects C. For formal definitions of F_{a,o_i} and D^C_{a,o_i} see Steels (1996b).

A *word* is a sequence of letters drawn from a finite shared alphabet. $W_a = (w_{a,1}..., w_{a,p})$ is the set of words known by agent a. Consonant-vowel sequences are used but the utterance serves only as an identifier (Steels 1996a). In the original guessing game described by Steels, an *expression* is a set of words; word order is not modelled.

A *lexicon* is a relation between possible feature set K and a word w. Each member of this relation is called an association. Each agent a is assumed to have a single lexicon L_a, which is initially empty.

A guessing game $l_g = \langle C, a_i, a_r, o_t \rangle$ consists of a context $C \subseteq O$, an initiator agent a_i, a recipient agent a_r and a topic object o_t. The game proceeds as follows:

1. The initiator determines a distinctive feature set for the topic ($D^C_{a_i,o_t}$). In the case of multiple possible distinctive feature sets, the smallest set is preferred, then the most abstract (the ones for which the sensory channels have the smallest set of features), then the set containing the most used features.
2. The initiator constructs an expression $e \in L_{a_i}$ which covers $D^C_{a_i,o_t}$. The $cover(D, L_a)$ function (formally defined in Steels 1996a) produces an expression such that all the features in D are expressed by words in e according to associations in L_a. In the case of multiple possible expressions, the same criteria as for selecting a set $D^C_{a_i,o_t}$ are used. In other words, the initiator chooses words that express its desired meaning.
3. The recipient uncovers from e the feature set $f \in F_{a_r}$. The $uncover(e, L_a)$ function (Steels 1996a) produces the set of features that are expressed by the words in e according to the associations in L_a. In other words, the recipient extracts its perceived meaning from the initiators message.
4. The recipient selects the set of objects O_f that are identified by the set f given the set of feature detectors Σ_{a_r}. $O_f = \{o | F_{a_r,o} \supseteq f\}$. An object $o_f \in O_f$ is chosen based on which feature detectors have been most successful in prior games.
5. The recipient non-verbally identifies of to the initiator, this is described as "pointing" (Steels 1996b). The language game l_g ends in communicative success if $o_f = o_t$.

The guessing game can end in failure if one of the following occurs:

1. $D^C_{a_i,o_t} = \emptyset$. There are not enough distinctions in S_{a_i} to identify o_t and therefore $\forall o_c \in C, F_{a_r,o_t} \subseteq F_{a_r,o_c}$. When this occurs, the agent will construct a new feature detector for an empty sensory channel if any exist. Otherwise an existing sensory channel will be refined by creating a new feature detector that segments an existing feature.
2. $cover(D^C_{a_i,o_t}, L_{a_i}) = \emptyset$. The initiator a_i may not know enough words to cover all the features. When this occurs, the agent will create a new word.
3. $uncover(e, L_{a_r}) = \emptyset$. The recipient may not have enough associations to uncover all the meanings in e. When this occurs, the agent will create a new meaning or adjust an existing one if a deduction can be made.
4. $o_f \neq o_t$. The object identified by the recipient is not the object selected by the initiator and therefore $uncover(e, L_{a_r}) \not\subset F_{a_r,o_t}$. In this event the success records for the involved associations are not incremented.

For detailed descriptions of the actions taken by the agents in the event of these failures see Steels (1996b).

The Generation Game

In the generation game, agents produce designs based on their understanding of the requests of other agents. This creates the basic principal of design: a designer must produce a design that exhibits a set of required behaviours according to a client. This mechanism is embedded within a language game between an initiator agent and a designer agent. The generation game is similar to the guessing game in that the agents communicate an expression to be identified, and similar to the imitation game in that the recipient must construct something that exhibits the properties conveyed as an expression.

The set of objects $O = (o_1, ..., o_m)$, the set of sensory channels $\Sigma = (\sigma_1, ..., \sigma_n)$, the set of feature detectors $S_a = (s_{a,1}, ..., s_{a,m})$, the features of an object F_{a,o_i}, words (w), expressions (e) and lexicons (L) are defined as in the guessing game described above. In addition, let there be a set of generative actions G_a possessed by agent a, such that a set of generative actions $G_{a,o_i} \subseteq G_a$ produces object o_i. The definition of a set of generative actions G_{a,o_i} is implementation specific.

A generation game $L_g = \langle a_i, a_d \rangle$ consists of an initiator agent a_i and a design agent a_d. The game proceeds as follows:

1. The initiator generates a "requirements" feature set r. The requirements feature set includes values for features that the initiator agent desires in the designed object.
2. The initiator constructs an expression $e \subseteq L_{a_i}$ which covers r. The $cover(r, L_a)$ function operates as described in the guessing game for the distinctive feature set D.
3. The designer uncovers from e the feature set $f \subseteq F_{a_d}$. The $uncover(e, L_a)$ function operates as described in the guessing game.
4. The designer produces $G_{a_d,o_d} \subseteq G_{a_d} | f \subseteq F_{a_d,o_d}$. In other words, the designer produces a set of design actions such that the feature set of the designed object o_d includes the features uncovered from the initiators expression f.
5. The designer displays o_d to the initiator. The generation game L_g ends in communicative success if $F_{a_i,o_d} \supseteq r$.

The generation game can end in failure if one of the following occurs:

1. $F_{a_i} = \emptyset$. The initiator possesses no features. If this occurs, the initiator will generate o using a random set of generative actions $G_{a_i,o}$ and create new feature detectors from it.
2. $cover(r, L_{a_i}) = \emptyset$. The initiator a_i may not know enough words to cover all the required features. When this occurs, the agent will create a new word.
3. $uncover(e, L_{a_d}) = \emptyset$. The designer may not have enough associations to uncover all the meanings in e. When this occurs, the agent will create a new meaning or adjust an existing one if a deduction can be made.
4. $r \not\subset F_{a_i,o_d}$. The object produced by the designer does not satisfy the design requirements produced by the initiator and therefore $uncover(e, L_{a_d}) \neq r$. In this event the success records for the involved associations are not incremented.

The most significant difference between the guessing game and the proposed generation game, is that the recipient agent in the guessing game must select the object o_f from the context C, whereas the designer agent in the generation game must generate the object o_d by some process. The details of the design process are domain specific and so are not specified in the language game itself. In addition, the generation of the requirements feature set, unlike the distinctive feature set of the guessing game, requires a process to be specified.

The model of interest, at the core of the computational model of curiosity, can be used in either of these generative process. Used by the initiator agent a_i an hedonic function will promote an exploration of the space of feature sets, such that the requirement feature sets produced are similar-but-different to those produced in the past. Similarly, if the designer agent a_d uses an hedonic function, it will motivate the search of the space of possible designs to locate novel designed objects.

The generation game emphasizes the distributed nature of creativity. The possibility of incorporating a model of curiosity in either the client or the designer illustrates how, in a typical interaction, both parties have an opportunity to be creative. As noted by Getzels & Csikszentmihalyi (1976), the activity of problem–finding can be just as creative as problem–solving.

An Example Domain

To assist in understanding how language games can be implemented in a domain, this section will examine the components of an implementation for the domain of Spirograph[1] patterns. The Spirograph is familiar example of a generative

---

[1] Spirograph is a registered trademark of Hasbro.

system that can produce complex patterns from the specification of a few parameters, the selection of a stationary wheel, a moving wheel, and a hole in the moving wheel for a pen to draw through.

A simple arrangement of circular gears, as illustrated in Figure 4, can be modeled mathematically using the following equation:

$$\begin{aligned} x &= (r_1 + r_2) \times \cos\theta_1 - p \times \cos\theta_2 \\ y &= (r_1 + r_2) \times \sin\theta_1 - p \times \sin\theta_2 \end{aligned}$$

where:

r_1 = radius of fixed gear
r_2 = radius of moving gear
p = distance of pen from centre of moving gear
θ_1 = rotation of moving gear around fixed gear
θ_2 = rotation of moving gear

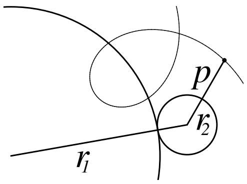
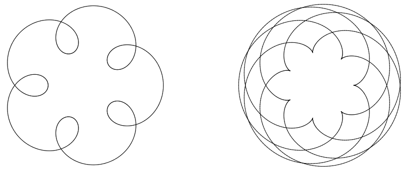


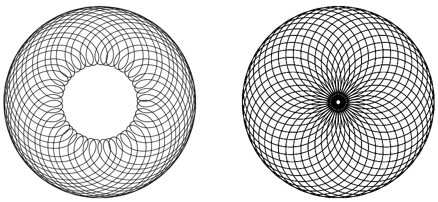
Figure 4: A simple Spirograph generator using two circular gears and an extended pen position.

Altering these parameters slightly can produce a range of different designs. As can be seen in Figure 5, these designs can differ in tone, size and texture.



(a) $r_1 = 200$, $r_2 = 40$, $p = -100$

(b) $r_1 = 140$, $r_2 = 100$, $p = 100$



(c) $r_1 = 200$, $r_2 = 78$, $p = -122$

(d) $r_1 = 200$, $r_2 = 5$, $p = 195$

Figure 5: Some example Spirograph patterns.

To apply the language games described above to the domain of Spirograph patterns, we must define a set of utterances, a set of design actions and a set of feature detectors. The set of utterances, used to construct words provides are not domain–specific and the set already defined for previous models of the evolution of language, e.g., `(k a)`, `(y u)`, `(z e)`, `(v o)`, etc.

Previously, curious design agents have been used to explore the space of Spirograph patterns and have showed considerable fixation on areas of the design space that contained unexpected patterns (Saunders & Gero 2001). To generate Spirograph patterns agents perform a guided walk by adjusting the parameters in small increments. Curious agents use the novelty of surrounding patterns to guide the search process, where novelty is based on the distance from a set of prototypes constructed from previously generated patterns.

To enable the use of language the agents must be able to discriminate between different Spirograph patterns. This is achieved through a set of functions that can be used by the agents to construct feature detectors as required. A useful set of functions can be constructed using simple measurements of the pixel values of images of the patterns produced as described in Table 1.

Table 1: Example feature detectors for Spirograph patterns

Feature	**Method of Sensing**
Contrast	Pixel value range
Brightness	Average greyscale pixel value
Symmetry	Difference of symmetrical pixels
Size	Radius of non-background pixels
Noise	Entropy of pixel image

Each function defines a sensory channel. Feature detectors are constructed on demand from these sensory channels on demand, as a consequence of a communicative failure. For these sensory channels, features are defined as ranges of values that these channels support. For example, given the sensory channel that measures size a number of feature detectors may be defined that define such features as “small” and “large”. As features are defined they are associated with initially randomly chosen words. After a number of language games a population of agents will come to an agreement, e.g., “small” = `(y u)` and “large” = `(k a)`. Ultimately, as the pressure to form more distinctive expressions increases, the language games will result in more complex expressions, e.g., “small dark” = `(y u)(v o)`.

Within a design space like the Spirograph, the two language games described above may be used to model different aspects of the role that the evolution of language may play in the development of a creative domain.

Modeling Design

The generation game, as presented above, with a client producing a set of requirements that a designer works to satisfy, is a simplification of a working creative design partnership but it serves to illustrate how the creative process is shared between different stake-holders.

The client generates a requirement feature set r, e.g., $\{\sigma_{size} = 100, \sigma_{brightness} = 0.1\}$, which represents a concept, e.g., “small dark”. The client then expresses the features as a set of words `(y u)(v o)` and utters this to the designer. The designer uncovers the meaning of this expression and designs according to its understanding of its meaning, which may be different from the clients depending on how well defined the concepts have become in the system e.g., $\{\sigma_{size} = 90, \sigma_{brightness} = 0.15, \sigma_{contrast} = 0.3\}$.

To produce the design, a curious design agent constructs an evaluation function using distance metrics appropriate to each sensory channel in the requirements, such that the evaluation function returns a high value for Spirograph patterns that match the required features. The agent then performs a search of the space of possible Spirograph patterns to find an "interesting" pattern that achieves a high score for the constructed evaluation function.

Like the previous models of creative fields, the addition of works into the repository of valued works is controlled by agents other than the designer. In the simplest form of interaction, if a client agrees that a designer has satisfied the requirements for a particular pattern in an interesting way, the design may be added by the client to the repository.

Modeling Education

The guessing game is capable of developing common agreement on the use of words over a population. As such it represents a language game that can be used to model forms of education in a creative domain, where the goal of the repeated application is to provide a common communicative framework. Education plays an important role in an individual's mastery of a domain. It is only by learning the history of valued works and the language used to describe them that an individual can hope to contribute something new and describe it in such a way as too have it accepted by it's field.

To model such an educative role, the initiator agent takes on the role of teacher, selects a context C and a topic o_t from the Spirograph patterns already present in the repository of valued works in the domain. A distinctive feature set is constructed within the context of the other valued objects in the domain $D^C_{a_i,o_t}$ and an expression is constructed e. The recipient agent, a_r, takes on the role of student and must attempt to identify the pattern being described by the initiator agent, given the context C.

Modeling Multiple Domains

One advantage of using simple pixel-based sensory channels like the ones described above is that they can potentially be used across multiple image-based design spaces, allowing language games to be played between compatible domains. Naturally, in domains that use very different generative systems, the lexicons will evolve differently to reflect the distinctive features of works produced, however, features shared across domains will result in overlapping lexicons that share some common words.

Discussion

It is certain that computational modeling will continue to focus on the developing analogs for creative cognition and individual creative behavior. After all, the promise of developing computer programs able to solve problems in ways that are obviously "creative" is so tantalizing that we cannot help ourselves. What this paper seeks to accomplish, however, is to show that the potential exists for developing computational models that capture how creativity works within a cultural environment.

The new language game presented here represents only our first attempt to model creative activity within a field that involves language. There are several different kinds of creative individual (Policastro and Gardner 1999) and each kind may take part in different types of language games as they interact. Exploring models of the evolution of language in creative domains opens up the potential to investigate a range of potentially important aspects of creativity that are outside the scope of studies focussed on individuals, for example:

- The effects of a common education on the production and evaluation of creative works
- The emergence of specialized languages that are grounded in the practices of a field
- The emergence of subdomains as a consequence of differences in language use across a field

The computational model presented here advances the computational modeling of the DIFI framework by introducing a way for language for the description of works to develop from the interactions within a creative system. Future work in this area will need to also need to incorporate similar mechanisms for the evolution of policies and rules.

References

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

Csikszentmihalyi, M. 1988. Society, culture and person: a systems view of creativity. In Sternberg, R. J., ed., *The Nature of Creativity: Contemporary Psychological Perspectives*. Cambridge, UK: Cambridge University Press. 325–339.

de Boer, B. 2000. Emergence of vowel systems through self-organisation. *AI Communications* 13:27–39.

Feldman, D. H.; Csikszentmihalyi, M.; and Gardner, H. 1994. *Changing the World: A Framework for the Study of Creativity*. Westport, CT: Praeger Publishers.

French, R. M. 2006. The dynamics of the computational modeling of analogy-making. In Fishwick, P., ed., *CRC Handbook of Dynamic Systems Modeling*. Boca Raton, FL: CRC Press LLC.

Gardner, H. 1993. *Creating Minds: An Anatomy Of Creativity As Seen Through The Lives Of Freud, Einstein, Picasso, Stravinsky, Eliot, Graham, and Gandhi*. New York, NY.: Basic Books.

Gemeinboeck, P., and Dong, A. 2006. Discourses of intervention: A language for art & science collaboration. In *Proceedings of International Conference: New Constellations: Art, Science and Society*.

Getzels, J. W., and Csikszentmihalyi, M. 1976. *The Creative Vision*. New York, NY.: Wiley.

Gruber, H. E. 1981. *Darwin on Man*. Chicago: University of Chicago Press.

Martindale, C. 1990. *The Clockwork Muse*. New York, NY.: Basic Books.

Miranda, E. R.; Kirby, S.; and Todd, P. M. 2003. On computational models of the evolution of music: From the origins of musical taste to the emergence of grammars. *Contemporary Music Review* 22(3):91–111.

Newell, A., and Simon, H. A. 1972. *Human Problem Solving*. New York, NY.: Oxford University Press.

Newell, A.; Shaw, J. C.; and Simon, H. A. 1958. Elements of a theory of human problem solving. *Psychological Review* 65(3):151–166.

Saunders, R., and Gero, J. S. 2001. A curious design agent: A computational model of novelty-seeking behaviour in design. In *Proceedings of of the Sixth Conference on Computer Aided Architectural Design Research in Asia (CAADRIA 2001)*, 345–350.

Saunders, R., and Gero, J. S. 2002. How to study artificial creativity. In *Proceedings of Creativity and Cognition 4*.

Steels, L. 1995. A self-organizing spatial vocabulary. *Artificial Life* 2(3):319–332.

Steels, L. 1996a. Perceptually grounded meaning creation. In Tokoro, M., ed., *ICMAS96*. AAAI Press.

Steels, L. 1996b. The spontaneous self-organization of an adaptive language. In Muggleton, S., ed., *Machine Intelligence 15*. Oxford University Press, Oxford.

Steels, L. 1998. The origins of syntax in visually grounded robotic agents. *Artificial Intelligence* 103(12):133–156.

Wittgenstein, L. 1953. *Philosophical Investigations*. Blackwell Publishing.