INTERNATIONAL CONFERENCE ON ENGINEERING DESIGN, ICED'09
24 - 27 AUGUST 2009, STANFORD UNIVERSITY, STANFORD, CA, USA

EXPLORING KNOWLEDGE TRANSFER IN A SOCIETY OF DESIGNERS

Andrés Gómez de Silva Garza[1] and John S. Gero[2]
(1) Instituto Tecnológico Autónomo de México (2) George Mason University

ABSTRACT

This paper presents an approach to understanding designing that includes societal effects. We use a multi-agent system to simulate the interactions between two types of agents, those that produce designs and those that receive the designs proposed by the producers. Interactions occur when receiver agents give their opinion to the producers on the designs they produced. Based on this information some producers may choose to adopt knowledge from other producers and some receivers may choose to adopt knowledge from other receivers. As a result of this exchange of knowledge global behaviors emerge in the society of agents. We provide the results of preliminary experiments with the model in which we measure variations in the producers' successes, which allows us to observe and analyze some of these emergent behaviors under different knowledge transfer scenarios.

Keywords: Society of designers, computational model, situated design cognition

1 INTRODUCTION

Most existing computational models of design propose algorithms that simulate the behaviors and generate the products of a single designer. These models are limited when it comes to being able to predict or simulate the societal effects on designers.

Design in the world occurs within a social context. There are two important aspects to this social context. One is that there are generally several designers producing competing or similar types of designs, not just one designer. For example, each major city in the world has many architectural firms operating in it, not just one, and within each firm there are several architects actively engaged in proposing designs, not just one.

The second important aspect of the social context of design is that the designs generated by each designer depend on the designer's experience, and that experience is informed by feedback from the world about previous designs, where the feedback is provided, either directly or indirectly, by other agents and their opinions on previous designs. Experience in this context is not so much a collection of knowledge that has been previously stored by the designer in its memory, as it is traditionally modeled, but rather emerges as a designer receives and integrates feedback from the world about designs he has previously created into the decisions he makes in the production of future designs. The type of feedback that can be helpful in this context is information about the quality, the usefulness, and the effectiveness, of previous designs. This type of information cannot be obtained or inferred directly by the designer in isolation, because it depends on the subjective opinions of the users and observers of the designs produced by the designer.

Designing is thus viewed as a social act, and designers change over time because their world view changes over time based on their experience. For example, in the initial stages of the development of personal computing in the 1980's, people were split up into two separate camps, the IBM-compatible users whose programs and files were fully exchangeable across platforms, and the Apple Macintosh users, who were not able to easily share programs and files with the users of any of the IBM-compatible computers. Over the years, Apple started losing market share to the IBM-compatible computers. In the 1990's Apple decided to make their computers more compatible with the IBM-compatible ones based on the negative feedback they had received from their users. If the behavior of the market (i.e., the response of the users of both types of computers and its evolution over time) had not occurred the way it did, there would have been no reason for Apple to promote IBM-compatibility as a design feature in their new models. Thus, design as an intellectual activity does not end when one design has been proposed by one designer in order to solve one need, but continues throughout a designer's career and is richly influenced by a designer's interaction with the world.

The need to consider an agent's environment and its interactions with other agents in general studies of cognition and in computational models of reasoning scenarios has been pointed out by several researchers ([1] and [2]) and when it includes the notion of the agent being affected by its view of the world is generally referred to as situatedness. Several studies have focused on design cognition while discussing situatedness ([3], [4], [5] and [6]). The unifying themes of these studies are the importance of the environment (in particular, other designers and receivers of designs) when a designer is making design decisions, the changing nature of the factors that influence those decisions, and the emergence of global behaviors in the societies of designers as a consequence of the individual decisions made by the designers.

In this paper we use these ideas to propose a computational model of the social aspects of design activity and we provide the results of some preliminary experiments with the model. This model is a multi-agent system that consists of design agents (that is, agents that produce designs), which we call *producers*, and non-design agents (that is, agents that do not produce designs but that observe and evaluate the designs created by the producers), which we call *receivers*. The model fits well within, and contributes a computational implementation of, the DIFI (Domain-Individual-Field-Interaction) framework proposed in [7] and shown in Figure 1, which views creativity as a property of the interaction between individuals in a society (field) that belong to a given culture (domain).

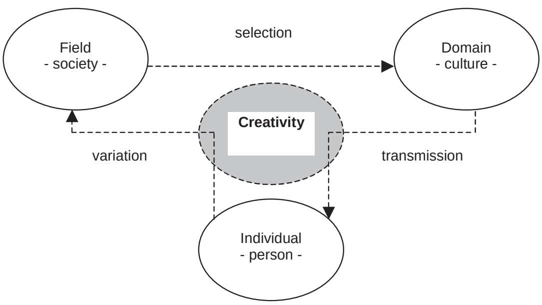

Figure 1. DIFI framework: creativity as a property of Domain-Individual-Field-Interaction (after [7])

2 KNOWLEDGE MODEL OF PRODUCER AND RECEIVER AGENTS, AND IMPLEMENTATION DOMAIN

Many types of algorithm have been proposed in the creation of computational models of design. One such type of algorithm is Genetic Algorithms (GA's, see [8]).

Each producer in the model uses a GA to evolve and eventually propose new designs. In a GA, the guiding force for how potential designs evolve is the GA's fitness evaluation function that is used in the algorithm to determine the quality of each new potential design that is generated. This fitness evaluation is what determines which potential designs survive through multiple generations and thus the general type of design which an agent's efforts tend to produce. Another important factor is the genetic alphabet used to initialize the GA's population of evolving potential designs, because the makeup of this alphabet guides and constrains the type of designs that an agent is capable of describing and thus producing.

The fitness evaluation functions used by a producer embody its goals and hence the type of designs it produces. The genetic alphabet used by a producer embodies the type of actions or decisions it can employ or make while producing designs. In the model it is these two sets of knowledge, the goals and the actions, that can be affected over time based on feedback from the world about the quality of previous designs produced.

The example design domain we have implemented is the production of shapes consisting of agglomerations of unit squares, where the squares can have one of four colors: white, red, green, or blue. This is an exemplary domain that can be readily substituted by more engineering domains. However, in these experiments this domain suffices to demonstrate the issues involved. The following two sub-sections describe the alphabet symbols and evaluation functions that we have implemented for this design domain.

2.1 Genetic alphabet of producers

A subset of the alphabet that is available to design agents in this domain, each symbol of which consists of making a move to a different position and placing a unit square of a particular color there, is shown graphically in Figure 2 (the full alphabet in the implementation consists of strings that represent the visual symbols shown in the figure). The universal alphabet repeats the same symbols shown in the figure for each of the four possible colors of the unit square placed in the end location, and so consists of a total of 32 symbols. A shape is described by concatenating several of these symbols and interpreting the end location of one symbol to be the start location of the following symbol in the sequence.

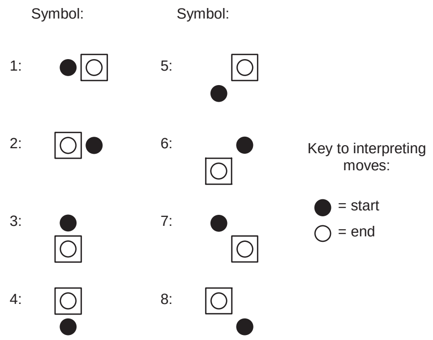

Figure 2. Subset (for a given color) of the genetic alphabet of producers

Some consequences of using such an alphabet are that:

- Shapes are described by the actions needed to produce them. This leads us to the distinction between the genotype (the actions) and the phenotype (the shape), and the need for a genotype-to-phenotype mapping
- Each move implies a change of location from a starting unit square to an ending unit square. A concatenation of symbols forming a genotype consists of a series of moves in which the end location of each move is the start location of the next move. The end location of each move is a unit square of a given color that is added to the shape being constructed when the genotype is decoded. If any location is re-visited several times during the decoding of a genotype then the unit square's final color in that location in the phenotype is the last color specified by carrying out the instructions in the genotype. The location in which the phenotype begins to be constructed by following the instructions in the genotype will not belong to the phenotype unless it is re-visited later on during genotype-to-phenotype mapping (since the first symbol of the genotype, like all others, implies first making a move from a start location and then adding a unit square at the end location of the move to the phenotype).
- The action of adding a unit square to the sequence of unit squares that describes a shape is not explicitly described in the genotype separately from the action of making a move to a different location, since it is assumed that each location (unit square) visited as a result of the actions (movements) in the genotype is meant to be added to the shape described by the genotype.
- There are several ways to describe the same shape. A row of two unit squares of a particular color can be described, using the numbers assigned to the symbols shown in Figure 2, as “11” (constructing the row by starting to its left), but the same row of two unit squares of the same color can also be described by “22” (constructing the row by starting to its right), which is

equally efficient, by “112” (starting to the left of the row and then back-tracking and re-visiting the first square placed), which is slightly less efficient, or by multiple other combinations of symbols from the alphabet. Thus, there is a many-to-one mapping from genotypes (sequences of actions) to phenotypes (shapes produced by following the sequences of actions).

2.2 Evaluation functions of producers and receivers

While the production of shapes is done only by producers, both producers (as part of the process of deciding which shapes to keep and which to discard during their use of a GA to evolve shapes) and receivers (as part of the process of observing and giving feedback on the shapes that producers have produced) use different criteria to assign a value to a given shape. These criteria are embodied in fitness evaluation functions. In order to be able to combine evaluation functions (for instance, if one agent wants to use two criteria simultaneously in order to evaluate a given design) or to compare the results of applying them (for instance, if one producer wants to know how its designs compare with those of other producers) we have normalized the values returned by all evaluation functions to fall between 0 and 1. These two extreme values can be interpreted as the shape being 0% "unsuccesful" or 100% "succesful" according to a criterion, respectively. The linear combination of several such values (adding them together and dividing by the total number of values that were added) will also result in a value that will fall between 0 and 1 and can be interpreted in a similar manner as the global fitness value of a shape, taking into account several criteria at a time (and giving them all the same weight/importance).

We have implemented evaluation functions, each of which focuses on a geometric aspect of the shape being evaluated or on some aspect of the coloring of the unit squares that the shape is made of, for the following properties:

- Genotype efficiency: A shape consisting of two unit squares side by side can be described in many ways as a sequence of moves, for example "11", "22", or "112", to list just a few. Genotype efficiency measures how economically the production of the shape is described, how close the actual number of genes is to the minimum possible number of genes required to describe a particular shape.
- Tallness of shapes.
- Flatness of shapes.
- Area-to-perimeter ratio of shapes.
- Bumpiness of shapes.
- Degree of convexity of shapes: The percentage of the angles in the perimeter of the shape that are non-reflex angles.
- Degree of concavity of shapes: The percentage of the angles in the perimeter of the shape that are reflex angles.
- Vertical symmetry of shapes.
- Horizontal symmetry of shapes.
- Degree of color saturation of shapes.
- Checkeredness of shapes: The checkeredness of a shape is a measure of how much alternation there is between the colors of its unit squares in the up-down and left-right directions.
- Contiguousness of colors in shapes: The less change of color there is by going from one unit square in a shape to another in the up-down and left-right directions, the more contiguous the colors that occur within the shape are.
- Diagonality of colors in shapes: The diagonality of colors in shapes (i.e., to what degree there exist lines of diagonally-adjacent unit squares of the same color within a shape) is measured by dividing the number of diagonal adjacencies of unit squares that exist within a shape between two unit squares that are of the same color by the total number of such adjacencies (irrespective of the squares' colors).
- Occurrence of patterns of colors in shapes: Patterns are measures of how many X-shaped, H-shaped, hollow square shaped, hollow diamond shaped, and non-hollow (2x2) square-shaped sub-shapes of a given color exist within a shape for each of the four colors in the alphabet.

3 MODEL OF THE SOCIETY OF PRODUCERS AND RECEIVERS

We model the society of producers and receivers as a social multi-agent system (see [9], [10], and [11]). In any society, not all agents, even if they all participate in the same kind of task or activity such as designing, will pursue the same goals or be capable of exhibiting the same skills. In the computational model of this society of agents we therefore have the universal sets of alphabet symbols and evaluation functions described in the previous section available, but each agent only has a subset of each of these sets forming part of its repertoire.

We subdivided the universal set of evaluation functions into two overlapping subsets, one for producers and one for receivers, containing 15 functions each. In the simulations we have a total of 1,000 agents. These have been subdivided so that there are 25 producers and 975 receivers based on the United States census data showing that 2.5% of the population is involved in design- or creativity-related activities [12]. We randomly initialize each producer to have 4 evaluation functions, chosen from the set of producer evaluation functions, and each receiver to have 4 evaluation functions, chosen from the set of receiver functions. We also randomly initialize each producer to have 8 alphabet symbols chosen from the set of 32 that are available. However, each agent's subset of evaluation functions (and, in the case of producers, each agent’s subset of alphabet symbols) changes over time based on feedback from the world, as detailed below. When the simulation begins, each producer uses a GA (whose population is initialized randomly) to produce designs according to its own set of fitness functions. The first time-step in the simulation ends after convergence has occurred in the GA's of all of the producers.

Before beginning the next time-step in the simulation (i.e., before the producers again use a GA to create the next round of new designs), feedback from the world is used to transfer knowledge between some agents. It is these interactions between agents that can cause some agents to influence other agents' future behavior.

Producers make their best designs available for public viewing after each time-step. When this happens, the receivers each evaluate these designs according to their own fitness evaluation criteria, thus assigning a degree of quality (fitness value), falling between 0 and 1, to each producer (actually to each producer’s design). This simulates the way in which, for example, different competing products in a catalog are ordered by the public based on their own personal criteria for evaluating them. Once this "voting procedure" has occurred, each receiver will have produced a value judgment of the quality of the design created by each producer. Each producer's average quality (as judged by the set of receivers) is calculated. Figure 3 shows the data structure that holds the resulting information, which assumes that we have *m* producers and *n* receivers in the simulation.

		Producers:				
		P0:	P1:	P2:	...	Pm:
Receivers	R0:	q00	q10	q20	...	qm0
	R1:	q01	q11	q21	...	qm1
	R2:	q02	q12	q22	...	qm2
	...	...	...	...	...	...
	Rn:	q0n	q1n	q2n	...	qmn
Average Quality of Producers:		Q0	Q1	Q2	...	Qm

Where:
Qj=(qj0+qj1+…+qjn)/n

Figure 3. Matrix of votes and vector of average quality values

The different types of knowledge transfer that have been implemented are:

- *Producers adopting fitness functions from other producers:* Let Qmax be the average quality of the producer that was awarded the highest average quality after the voting procedure was performed (i.e., Qmax is the highest value in Q0, Q1, Q2, …, Qm) and Qmin be the lowest average quality value in the list. The most successful producer, the one that achieved an average quality value of Qmax after the voting procedure, used a certain set of fitness functions to evaluate alternatives when creating its designs. A certain number of the least successful producers (again, found by consulting the vector of average quality values for the producers) can adopt one of the fitness functions employed by the most successful producer, chosen at random. The transfer of knowledge in this case involves replacing one of the fitness functions formerly used (in the previous time-step) by the not-very-successful producer with the one adopted from the most successful producer. The least successful producers that adopt a fitness function from the most successful one are those that were awarded average quality values below a certain threshold value, calculated as a percentage (currently set to 2.5%) of the range of values between Qmin and Qmax. This type of knowledge transfer models the way in which producers that were not able to compete as well as some of their colleagues may adopt some of the design goals (embodied as fitness functions) of their more successful colleagues.
- *Producers adopting design ideas from other producers:* In an analogous fashion to the transfer of fitness functions from the most successful producer to several of the least successful ones, we can also transfer a gene from the most successful producer to several of the least successful ones. Again, which gene to transfer from the most successful producer and which gene used in the previous time-step by the least successful producers to replace is decided at random, and the simulation is currently set to do it for the producers whose average quality value fell below 2.5% of the range of values between Qmin and Qmax. This models the way in which producers that are not able to compete as well as some of their colleagues may adopt some of the design ideas (embodied as genes) performed by their more successful colleagues.
- *Producers adopting genotypes from other producers:* While receivers have no direct access to or knowledge of the way in which producers create their designs (that is, receivers can observe designs (phenotypes) but do not know the design ideas (genotypes) from which those designs were constructed), other producers often have at least a partial understanding about how their colleagues go about creating designs. Whether this knowledge of other producers' design ideas is gathered in the real world through industrial espionage, licensing of technology, generalization based on personal experience, or reverse engineering of their designs, in the end it amounts to producers having at least partial access to the design ideas of other producers. We model this in the simulations by allowing the transfer of genotypes from the most successful producers to the least successful producers. Genotype transfer occurs by using the genotypes of the most successful producers to partially seed the initial population of the GA's of the least successful producers. In this situation it is not just the most successful, but several of the most successful producers, that become sources of knowledge that is transferred to several of the least successful ones. Which of the most successful producers will provide the genotype of their best design to which of the least successful ones can be adjusted independently, and they are currently set to those whose average quality values fall above 97.5% and those whose average quality values fall below 2.5% of the range of values between Qmin and Qmax, respectively.
- *Receivers adopting fitness functions from other receivers:* In an analogous fashion to the transfer of fitness functions from the most successful producer to several of the least successful ones, we also transfer fitness functions from the most enthusiastic receiver to several of the least enthusiastic ones. The most enthusiastic receiver is determined by finding which receiver assigned the highest fitness value to the producer that was awarded the highest average fitness value Qmax. The least enthusiastic receivers are determined as a percentage of those that assigned the lowest fitness values to the producer that was awarded the lowest average fitness value Qmin. Again, which fitness function to transfer from the most enthusiastic receiver and which fitness function used in the previous time-step by the least enthusiastic ones to replace is decided randomly. Again, the decision of which of the least enthusiastic receivers will adopt is currently set to those whose average enthusiasm fall below

2.5% of the range of values between Emin and Emax (calculated analogously to Qmin and Qmax but on the basis of qmin0, qmin1, qmin2, …, qminn). This models the way in which receivers (observers) of the creations of producers can infect other receivers with their enthusiasm by explaining their reasons (embodied in fitness evaluation criteria) for being enthusiastic. This is a word-of-mouth type of knowledge transfer.

4 EXPERIMENTAL SETUP AND RESULTS

The hypothesis is that the computational simulation described above is sufficient to produce emergent social behaviors that will differ depending on the knowledge that is transferred between the agents as time progresses. In order to test this hypothesis we set up the following experiments. The simulations are run for 1,000 time-steps for each of the ten knowledge transfer combinations shown in Table 1.

Table 1. Types of knowledge transferred in the ten scenarios tested

	Types of Knowledge			
Scenarios	*Producer Evaluation Functions*	*Producer Alphabet Symbols*	*Producer Genotypes*	*Receiver Evaluation Functions*
a) No Knowledge Transfer				
b) Only Producer Evaluation Functions Transferred	X			
c) Only Producer Alphabet Symbols Transferred		X		
d) Only Producer Genotypes Transferred			X	
e) Only Receiver Evaluations Transferred				X
f) Only Producer Evaluation Functions Not Transferred		X	X	X
g) Only Producer Alphabet Symbols Not Transferred	X		X	X
h) Only Producer Genotypes Not Transferred	X	X		X
i) Only Receiver Evaluation Functions Not Transferred	X	X	X	
j) All Four Types of Knowledge Transfer Performed	X	X	X	X

For each scenario we measured several variables at each time-step, one of which is the acceptability of producers, measured according to how many adopters the producers had during a given time-step. In this paper acceptability is equated with popularity. It is considered that a receiver is an adopter of a product of a producer if the receiver awarded that product the highest value out of the set of votes that that receiver assessed for all of the producers. Figure 4 shows the graphs of variance in the popularity of the producers in the population across time-steps for each of the ten combinations of knowledge transfer tested. The variance of the popularity is a measure of the spread of designs produced. A change in variance of popularity represents a socially-induced change of either the receivers, the producers or both.

A low variance in the popularity of producers implies that most producers are approximately equally popular. Similarly, a high variance implies a widely varying degree of popularity amongst the producers. Figure 4 shows that in a given scenario there tend to be many oscillations in the variance of the popularity of producers. That is, it is hard to predict the future distribution of popularity of the producers even if we know the current distribution. This could be due to the partially random nature of the knowledge transfer. While knowledge is transferred from the most successful producer and/or most enthusiastic receiver, the decision about which evaluation function(s) and/or alphabet symbol are transferred to the least successful producer(s) and/or least enthusiastic receiver(s) is made randomly as the knowledge which makes a producer successful is not known to the other group. Thus, there is no assurance that it is the knowledge most responsible for the agent's success or enthusiasm that is transferred (assuming that there is any one item of knowledge that is most responsible rather than the combination as a whole being what creates the higher degree of success or enthusiasm). Future work will explore whether agents can perform some analysis that will allow them to make an informed decision on which knowledge to adopt, rather than choosing it randomly, and on allowing the adoption of more than one item, and seeing how this affects the results.

However, if we look at Figure 4 in more detail, we can see that even scenario *a* (no knowledge transfer) has a graph with many oscillations, so these oscillations cannot be caused entirely by the partially-random mechanism being used in deciding which item of knowledge to adopt during knowledge transfer. It is possible that one factor causing such changes in producers' relative levels of popularity is the random nature of the genetic operators used in the producers' GA's. While the designs created by the producers in one time-step may be highly attuned to the evaluation functions being used by the set of receivers, because of the random aspect of the genetic operators there is no guarantee that the designs created in the next time-step will be similar to the previously-created ones in ways that will make the designs continue to be so attuned (even if the producers and receivers don't change their set of evaluation functions and/or alphabet symbols from one time-step to the next). We can see this by tracing the ranking (assigning a rank of 1 to the most favorably evaluated producer and 25 to the least favorably evaluated, after taking into account the voting of all the receivers) of one producer throughout the simulation to observe whether in fact it varies much. Figure 5 shows the rank of the producer that was judged to be the favorite of the set of receivers after the first time-step was completed under scenario *a* (the most "stable," without any knowledge transfer). As can be seen in Figure 5, even when no knowledge transfer is performed and even though this producer was in general ranked quite well, there are frequent, and often quite large, changes in the ranking of a given producer throughout the simulation (reaching a low of 19^{th} place on two occasions, in this case). If the producers had used a different type of algorithm to create designs instead of GA's, one not relying so much on random choice, this aspect of the system's behavior in the experiments would probably also be different.

The oscillations in the variance of the popularity of the producers in scenario *a* (no knowledge transfer), Figure 4, occur around a mean value of approximately 1500. We can consider this the "base" value because it is what happens when no knowledge is transferred between agents at any time-step. Scenario *c* (transferring only producer alphabet symbols) behaves similarly, and scenarios *d* and *f* (transferring only producer genotypes and receiver evaluation functions, respectively) oscillate around a mean value not too much lower than 1500. On the other hand, scenario *b* (only producer evaluation functions transferred), after a period of instability that lasts for approximately 200 time-steps, settles down to oscillating around a mean value that falls below 1000. This would seem to indicate that, despite not eliminating the unpredictability of the simulation, transferring evaluation functions between producers seems to reduce the variability in the popularity of producers somewhat.

The last few graphs in Figure 4 look different from the rest. They show that, under their knowledge transfer scenarios, the variance in the popularity of producers decreases exponentially, stabilizing after 200-250 time-steps, at which point the frequency (and in most cases the magnitude) of the oscillations in the variance are much reduced in comparison to those in the first few graphs, discussed above. That is, the simulations in these scenarios converge to a state in which most producers end up being judged similarly to the rest by the set of receivers, and things remain that way for the rest of the time-steps. This occurs in scenarios *g* (only producer alphabet symbols not transferred), *h* (only producer genotypes not transferred), and *i* (only receiver evaluation functions not transferred). What this indicates is that eliminating any of these three types of knowledge transfer is enough to stabilize the overall behavior (in the long run, after 200+ time-steps) of the simulation. However, the same phenomenon can be observed in scenario *j* (all four types of knowledge transfer are performed), so that *not* eliminating *any* of these three types of knowledge transfer results in similar overall behavior.

To sum up, the hypothesis that different types of social behaviors would emerge under different knowledge transfer conditions is supported by these simulations. None of the four types of knowledge transfer by itself led to qualitatively different social behavior compared to the situation we had in the scenario without knowledge transfer (though transferring only producer evaluation functions did have an effect on the magnitude of the oscillations in the variance of the popularity of producers). When we performed three types of knowledge transfer, however, we obtained a qualitatively different type of behavior, one in which, after an initial transition period in which the randomly-initialized knowledge used by the agents was altered, all producers ended up being evaluated equally by the set of receivers, and this remained stable over time, but this happened only if one of the three types of knowledge transfer that was performed was the transfer of producer evaluation functions.

Key to Experimental Conditions:	
a=No Knowledge Transfer b=Only Producer Evaluation Functions Transferred c=Only Producer Alphabet Symbols Transferred d=Only Producer Genotypes Transferred e=Only Receiver Evaluation Functions Transferred	f=Only Producer Evaluation Functions Not Transferred g=Only Producer Alphabet Symbols Not Transferred h=Only Producer Genotypes Not Transferred i=Only Receiver Evaluation Functions Not Transferred j=All Four Types of Knowledge Transfer

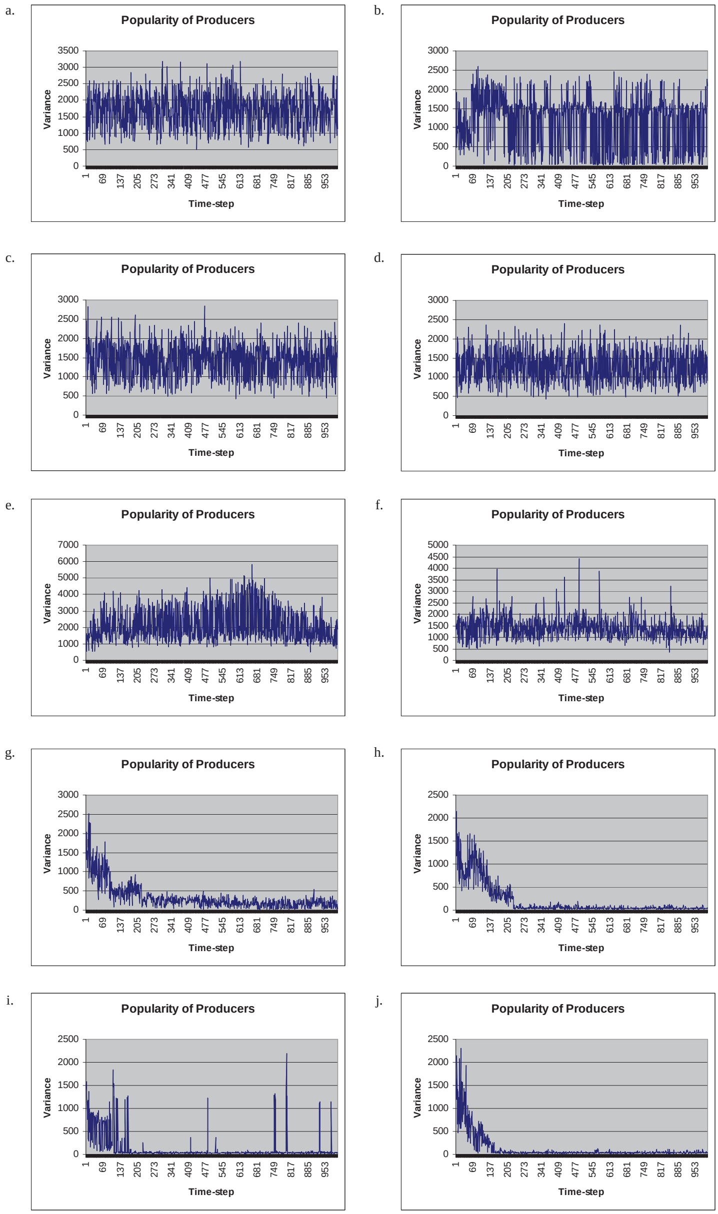

Figure 4. Graphs of variance in popularity of producers for ten knowledge transfer scenarios

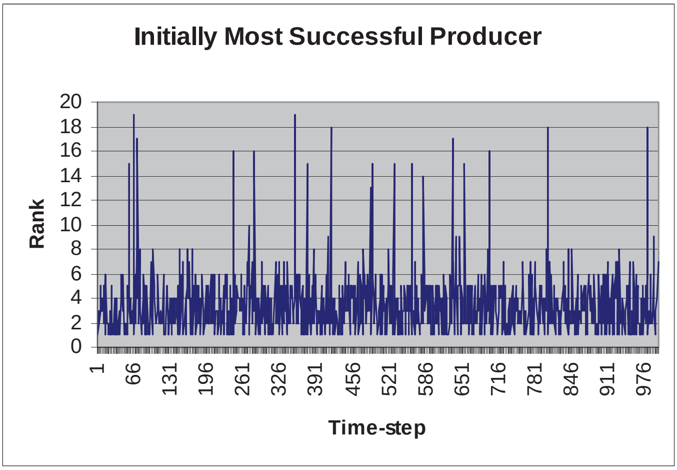

Figure 5. Rank of initially most successful producer when no knowledge transfer is performed

When we performed all four types of knowledge transfer, the same phenomenon occurred, which makes sense, since this includes the transfer of producer evaluation functions.

So we now have a secondary hypothesis which is that the absence or presence of the transfer of evaluation functions between producers is what causes the qualitative differences in the overall behavior of the simulation. In order to determine the validity of this hypothesis, we tested the remaining six possible combinations of knowledge transfer that we hadn't tried out during the initial experiment (three combinations of producer evaluation functions being transferred together with one other type of knowledge and three combinations of two types of knowledge that do not include producer evaluation functions being transferred), shown in Table 2.

Table 2. Types of knowledge transferred in the additional six scenarios tested

	Types of Knowledge			
Scenarios	*Producer Evaluation Functions*	*Producer Alphabet Symbols*	*Producer Genotypes*	*Receiver Evaluation Functions*
k) Producer and Receiver Evaluation Functions Transferred	X			X
l) Producer Evaluation Functions and Genotypes Transferred	X		X	
m) Producer Evaluation Functions and Alphabet Symbols Transferred	X	X		
n) Producer Genotypes and Receiver Evaluation Functions Transferred			X	X
o) Producer Alphabet Symbols and Genotypes Transferred		X	X	
p) Producer Alphabet Symbols and Receiver Evaluation Functions Transferred		X		X

Figure 6 shows the graphs of the variance in the popularity of producers under these six combinations.

Key to Experimental Conditions:	
k=Producer and Receiver Evaluation Functions Transferred l=Producer Evaluation Functions and Genotypes Transferred m=Producer Evaluation Functions and Alphabet Symbols Transferred	n=Producer Genotypes and Receiver Evaluation Functions Transferred o=Producer Alphabet Symbols and Genotypes Transferred p=Producer Alphabet Symbols and Receiver Evaluation Functions Transferred

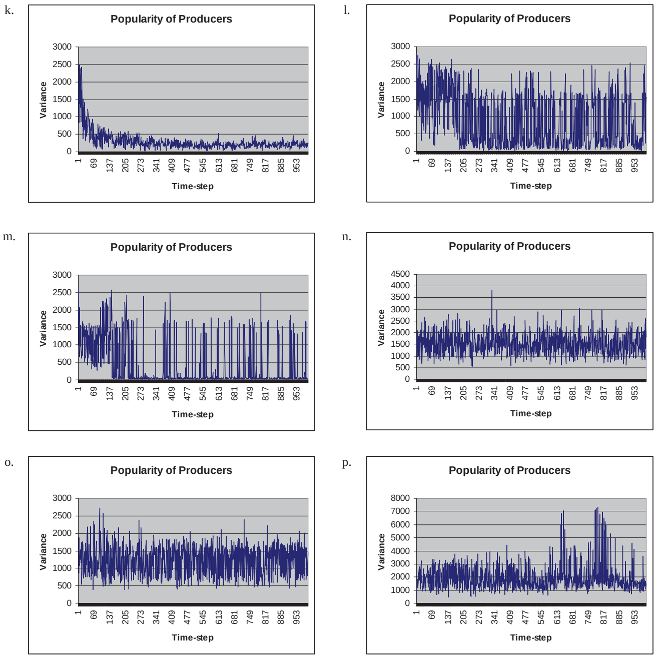

Figure 6. Graphs of variance in popularity of producers for a further six knowledge transfer scenarios

The three new knowledge transfer scenarios in which there was transfer of evaluation functions between producers, scenarios *k*, *l*, and *m* in Figure 6, in general all have a decreasing exponential shape (two with a large amount of oscillation in the variance being measured). This confirms the secondary hypothesis that the absence or presence of the transfer of evaluation functions between producers is what causes a qualitative difference in the overall behavior of the system. Of the three decreasing exponentials shown in Figure 6, scenario *k*, in which only evaluation functions (of both producers and receivers) are transferred, has the lowest oscillations, which implies that the transfer of evaluation functions in general, whether of producers or receivers, has a different effect on the behavior of the society than that of transferring other types of knowledge. In the future we plan to study the producer evaluation functions in more detail to determine what it is about them that seems to make them more important than the other types of knowledge in determining the overall behavior of this society of agents.

5 DISCUSSION

We have modeled a society of interacting agents, some of which produce designs and some of which receive them, by having the receivers make value judgments on the designs created by the producers and using the results of these judgments to influence the transfer of knowledge among producers and among receivers between each time-step in the simulation. The distinction between producers and receivers in the world is not precise since it is not just receivers, but also most producers, that critique the productions of other producers, and this influences the other producers' future productions. Separating the two functions conceptually into different types of agent permits us to more easily disentangle the complex interactions that occur between agents in a society and the influence of these interactions on future behavioral changes in the agents.

What we observed based on these experiments is that, on the basis of simple behavioral modifications to the individual agents based on transferring different types of knowledge, global behaviors of the society as a whole can emerge over time. One such behavior, eventual stability despite initial variation in the popularity of producers, seems to require that several types of knowledge be transferred and specifically seems to be connected with the transfer of goal-related knowledge between the producers in the shape of evaluation functions. According to these experiments, as long as this is one of the types of knowledge that is transferred, such stability emerges, whether or not there occurs any transfer of goal-related knowledge in the form of evaluation functions between receivers.

Computational modeling of social agents provides a laboratory to explore the implications of designers working within a social environment. The simple model described in this paper provides the foundation for future developments that can model emergent behavior between designers to form design "schools", the effect of location on exposure to designs and hence the potential to evaluate them, and the effect of innovation policies on the production of highly evaluated designs.

ACKNOWLEDGMENTS

This research is supported by a grant from the US National Science Foundation, grant no. CNS-0745390.

REFERENCES

[1] Clancey, W.J. *Situated Cognition: on Human Knowledge and Computer Representations*, 1997 (Cambridge University Press, Cambridge, U.K.).

[2] Ross, L. and Nisbett, R.E. *The Person and the Situation: Perspectives of Social Psychology*, 1991 (Temple University Press, Philadelphia, Pennsylvania).

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

[4] Love, T. Design as a Social Process: Bodies, Brains and Social Aspects of Designing. *Journal of Design Research*, 2003, 3(1).

[5] Sosa, R. and Gero, J.S. A Computational Study of Creativity in Design: the Role of Society. *Artificial Intelligence in Engineering Design, Analysis and Manufacturing (AIEDAM)*, 2005, 19(4), pp. 229-244.

[6] Smith, G. and Gero, J.S. What Does an Agent Mean by Being "Situated"? Design Studies, 2005, 26(5), pp. 535-561.

[7] Feldman, D.H., Csikszentmihalyi, M., and Gardner, H. *Changing the World: A Framework for the Study of Creativity*, 1994 (Praeger, Westport, Connecticut).

[8] Mitchell, M. *An Introduction to Genetic Algorithms*, 1998 (MIT Press, Cambridge, Massachusetts).

[9] Epstein, J.M. Generative Social Science: Studies in Agent-Based Computational Modeling, 2007 (Princeton University Press, Princeton, New Jersey).

[10] Epstein, J.M. and Axtell, R. Growing Artificial Societies: Social Science from the Bottom Up, 1996 (Brookings Institution Press, Washington, D.C.).

[11] Gilbert, G.N. and Conte, R. Artificial Societies: the Computer Simulation of Social Life, 1995 (UCL Press, London, U.K.).

[12] United States Census Bureau. *Census 2000 Indexes of Industry and Occupations*, 2000 (United States Census Bureau, Washington, D.C.).