A Computational Model of Collaborative Creativity: A Meta-Design Approach

Amit Banerjee, Penn State Harrisburg, USA

Juan C. Quiroz, SilverSky Group LLC, USA

Sushil J. Louis, University of Nevada, USA

ABSTRACT

The role of collaboration in the realm of social creativity has been the focus of cutting edge research in design studies. In this paper, the authors investigate the role of collaboration in the process of creative design and propose a computational model of creativity based on the newly proposed meta-design approach. Meta-design is a unique participatory approach to design that deals with opening up of design solution spaces, and is aimed at creating a viable social platform for collaborative design. A meta-design-based collaborative approach to the design process may achieve ET-creativity by expanding the conceptual space of design beyond what would have been possible by individual, non-collaborative design. The model has been implemented using interactive genetic algorithms, which casts the design problem as an optimization problem and uses a set of collaborative users for subjective fitness evaluation. The design problems investigated include the collaborative design of architectural floorplans and editorial design of brochures.

Keywords: *Artificial Intelligence, Computational Creativity, ET-Creativity, Floorplanning, Interactive Genetic Algorithms, Knowledge Creation, Meta-Design*

INTRODUCTION

Recent advances in computational creativity can be broadly classified into two groups – studies involving development of programs and computational techniques that are capable of intelligent and creative tasks, and those involving computer systems and programs that put the human-user in the loop with the aim of enhancing user creativity. The latter is often labeled computer-supported creativity. The definitions of creativity and creative design have been debatable, although it is broadly agreed that creativity is the generation of ideas that are both novel and valuable (Boden, 1999). The word "ideas" is domain-specific and has been used to mean concepts, products, processes, theories, melodies, paintings, numerous other forms of art, and so on. Novelty of ideas is defined with respect to past ideas, using either the creator or the entire humanity as a reference and keeping in mind the domain in which the idea is proposed, leading to classification of creativity into P-creativity (personal creativity)

DOI: 10.4018/jkss.2011040105

and H-creativity (historical creativity). Value to the idea can either be attributed at the time of proposition or later. In fact, many creative ideas have been recognized as being creative long after their being proposed – a result of the fact that creative ideas are ahead of their times and therefore can be impractical. An example of a creative idea in the business domain is the Netflix model of visual media distribution that is both novel (the model was far removed from static late-fee based Blockbuster model) and valuable (the implementation of the model made the parent company one of the most profitable technology companies in the last decade).

In this paper, we propose a new paradigm of computational creativity that is at the intersection of the two broad classes – a model that supports user-centric creative endeavors in design and is capable of producing designs that have creative aspects. The rest of the paper is organized as follows: in the next section we present the need for the new paradigm and review the formalized notion of computational creativity and the meta-design approach based on open systems. We then present the proposed collaborative interactive framework for creative design and relate it to the formalized notions of creativity and the meta-design approach. Before concluding we also present the algorithmic implementations of the model for collaborative design of architectural floorplans and editorial design of documents and brochures.

COMPUTATIONAL CREATIVITY: A FORMALIZED NOTION

Researchers in computational creativity are interested in the underlying process of creative ideation. This fundamental question has led to the definition of a conceptual space of ideas, which a computer program can search in. Creativity that is a result of simply searching the conceptual space for complete or partial possibilities is labeled E-creativity, for exploratory creativity. If the conceptual space is considered bounded by static rules, such E-creativity is often regarded as merely "innovation", and not creativity. On the other hand, if the rules that bound the conceptual space can be changed with respect to time, then the search for ideas in a continuously changing space is called T-creativity, for transformational creativity (Boden, 1999). This can be mapped to how a human thinker comes up with creative ideas. The mind is a veritable storehouse of ideas and if the mind could be mapped, this storehouse then becomes the conceptual space of ideas. If the thinker does not broaden his mind (by incorporating more domain knowledge, knowledge from other domains, etc.) he or she is just exploring a well-defined unchanging conceptual space producing ideas that may not necessarily be creative. It is only when the thinker moves out of rigid definitions of what-is and what-can-be, thereby modifying the conceptual space of ideas, that creative ideas are born. Kekule's discovery of Benzene rings, Watson and Crick's double helix model of the DNA, Frank Lloyd Wright's Fallingwater are just three of the many instances where thinkers and conceptualizers broke out of the well-defined mold and produced creative ideas. While what makes some people creative thinkers is a topic of ongoing neuroscience and psychology research, it can be presumed that T-creativity (breaking out of norms and boundaries) requires both an innate ability, varied knowledge and an applicative bent-of-mind.

A thinker may initially restrict the conceptual space of ideas intentionally to arrive at a first set of solutions and to develop an understanding of the problem domain and the solution space. The thinker can restrict the conceptual space based on the demanded requirements for the generated solutions, based on personal experience and intuition, based on solutions generated for similar problems, or based on common sense. However, once the initial conceptual space has been explored, it is up to the thinker to expand and change the conceptual space to produce creative ideas. Since an initial conceptual space can be expanded in many different ways, it follows that a first thinker may generate solutions with different creative qualities than a second thinker also generating solutions, even if both the first and the second thinker start from the same initial conceptual space.

In the computational creativity domain it has been argued that T-creativity is hard to achieve because systems that are able to spontaneously transform themselves (beyond what the system designer intended it to) are a philosophical oddity. Computational systems that produce artifacts deemed creative have been largely based evolution of random or human-proposed seed concepts. However, Boden judges a system by evolved artifacts for their creative content and proposes that a computational system exhibits T-creativity if such artifacts do not in any way resemble the seed concepts. Bill Latham's evolutionary art of 3D shapes (Todd & Latham, 1994) is compared to Karl Sims's 2D art program (Sims, 1991), both built on interactive genetic algorithms (which we will delve deeper in later sections) and both evolving attractive colored images and shapes from seemingly random seed images (and shapes). Since Sims's program produces images that are deemed more attractive by audiences (and are seen as bearing no resemblance to the seed images), his system was stated as having achieved T-creativity. Latham's 3D shapes seemed more or less similar to the starting forms, so it is surmised that Latham's program was just performing an evolutionary search in a fixed 3D shape-space, an example of E-creativity. This definition however precludes any engineering or product design system from ever achieving T-creativity, e.g. a computer system used to design kitchen layouts will always produce layouts that resemble a kitchen at some level.

Wiggins (2001) formalizes the notion of E-creativity and T-creativity in an attempt to address computational system creativity. He defines a global universe of ideas *U* that consists of finite subspaces called conceptual spaces *C*. *U* is a multidimensional space, whose dimensions are capable of representing anything; all possible mutually distinct ideas correspond with distinct points in *U*. The notion of the universe of ideas *U* is required because if *U* and *C* were the same and since each point in *U* can be reached by exploration, as a result only E-creativity would be possible. The conceptual space *C* is bounded by knowledge, experience and system limitations, and can be defined by a set of rules $\tilde{R}$. Members of *C* are chosen from *U* by implementing an interpretation $\left|\,.\,\right|$ on the set of rules $\tilde{R}$ as, $C = \left|\tilde{R}\right| U$. A different set of rules $\hat{R}$ is needed to devise a strategy to explore the conceptual space *C*. Exploration is defined as idea evolution – one can move from an existing idea c_0 to a new idea c_1 (c_0, $c_1 \in C$) using an interpretation $\left\|\,.\,\right\|$ on the set of traversal rules $\hat{R}$ as, $c_1 = \left\|\hat{R} \cup \tilde{R}\right\| c_0$. The bounding rules $\tilde{R}$ are needed to ensure that $c_1 \in C$. A third set of rules $\bar{R}$ is defined that allows the program to evaluate the quality of a potential concept. All three sets of rules $\tilde{R}$, $\hat{R}$ and $\bar{R}$ are expressed using a common language ℓ; the interpretations are also defined so as to interpret rules expressed using ℓ . E-creativity can be formalized using the septuple,

$$\left\langle U, \ell, \left|\,.\,\right|, \left\|\,.\,\right\| \tilde{R}, \hat{R}, \bar{R} \right\rangle$$

and occurs when a conceptual space *C* defined by $\tilde{R}$ is explored using well-written traversal rules $\hat{R}$ and evaluation rules $\bar{R}$. On the other hand, T-creativity happens when the set of rules $\tilde{R}$ can be modified from an existing $\tilde{R}_0$ to a new $\tilde{R}_1$ using transformation rules that ensure $\tilde{R}_1 \subset \ell$. However, Wiggins presents three scenarios that tend to weaken the contention that T-creativity is always superior to E-creativity. These are summarized:

1. The choice of traversal rules $\hat{R}$ might make it impossible to search for a particular concept *c* unless the traversal rules $\hat{R}$ are transformed. So although this is not akin to T-creativity, a transformation of traversal rules from $\hat{R}_0$ to $\hat{R}_1$ that lets the search process discover *c* is significant.

2. If the fitness landscape within a conceptual space is biased in a way to favor a certain choice of traversal rules $\hat{R}$ over others, a search process that uses the favored set of rules will uncover concepts in the convoluted regions of the fitness landscape and can be considered a creative search process.
3. A transformation of conceptual space C_0 to C_1 by transforming the bounding rules from $\tilde{R}_0$ to $\tilde{R}_1$ such that a new concept $c \in C_1$ is discovered. However if $C_1 = C_0 + \{c\}$, it is really hard to make the case that the T-creativity process that made the discovery possible is significant.

The model of computational creativity proposed in this paper builds on the first two points that Wiggins makes. We also extend Wiggins's formalized approach by proposing that by modifying traversal rules in continuously transforming conceptual spaces with an ever-changing fitness landscape, we achieve ET-creativity in the design process.

Collaboration and Meta-Design

A computational approach that supports T-creativity is the recently developed meta-design framework (Giaccardi & Fischer, 2008). Meta-design is a participatory approach to design that opens up solution spaces rather than let users manipulate complete solutions (as is the case of modifying artifacts in Sims's or Latham's visual systems). The intent is to allow the user to have control over the set of rules $\tilde{R}$, $\hat{R}$, $\bar{R}$ and thereby interact with the system in ℓ. The meta-design approach envisions designers as reflexive practitioners, who gradually build their understanding of the design problem as the design process progresses. We define design as a purposeful activity of devising new structures characterized by new parameters, aimed at satisfying certain requirements. Activities such as creating a business model, a product, an architectural layout, urban planning, human-computer interfaces etc., are all design activities in different domains. A creative support system is defined as one that allows the emergence of design possibilities with time, especially for ill-defined and complex design problems involving designers from different knowledge domains. Giaccardi and Fischer (2008) define open support systems as those that can be modified by designers and can evolve at use time, supporting more complex interactions. Use time is defined as the time when the support system interacts with the designer in the process of design. For example, imagine a computer support system assisting an interior designer in the process of developing creative kitchen layouts. The process of developing kitchen layouts is the process of design and happens at use-time. Build time on the other hand, is defined as the time when the support system itself was built by system and computer engineers. In order to qualify as open systems, support systems cannot be completely built prior to use and must be designed for evolution at the hands of the designer in a distributed manner. This imposes a practical limitation on how to build such open support systems – only a truly AI-based system will therefore be an open system.

In this paper we overcome this limitation by having the system evolve through collaboration between already-built remotely located environments. We hypothesize that collaborative interaction between peers working on a design problem at use time can exchange system information thereby altering how their own environment perceives the design process. This indeed relates to the proposition that to support open-ended and creative evolution it is fundamental for a user to be a part of the environment experienced by other users (Arthur, 1994; Taylor et al., 2002). The open systems envisioned by meta-design link creativity and evolution in that they "(1) promote the transcendence of the individual mind by supporting the differences in knowledge, abilities and motivations that exist among users, (2) support sustained participation by facilitating users' engagement in personally meaningful activities, and (3) enable the mutual adaptation and continuous evolution of users and systems by allowing users to evolve new ways of interacting with the environment and enabling systems to adapt to users' changing needs and practices"(Giaccardi & Fischer, 2008).

Collaborative systems have been the focus of studies into creativity and computer-supported cooperative work (Wilson, 1991) since the early 1990s. There has been a paradigm shift from computer-aided design systems to computer-supported collaborative design systems (Peng, 2001). It has been argued that much of our intelligence and creativity results from interaction and collaboration with other individuals (Csikszentmihalyi, 1997). Co-creation of concepts is one of the objectives included in the meta-design schema. Mediators used to facilitate users' engagement in the co-creation of meaningful concepts, are classes of environment excitations dynamically generated over the course of interaction by the interplay between the capabilities of the system and the individual concepts the users produce during the process of interaction. Mediators provide a concept for the emergence of meaningful concepts during the design process. Olivier Auber's Poietic Generator (Giaccardi & Fischer, 2008) has been referred to as a mediator-based system for co-creation of artistic images which allows different users to create space configurations on a shared canvas, the sum total of which either facilitates or inhibits the emergence of meaningful images on the canvas.

Affective mechanisms support co-creation by providing the environment for mutual interaction of different users with the support system. Affective mechanisms defined generally for creative digital arts enable users to experience the temporal and spatial features of the environment in terms of intentionality and proximity, e.g. they provide interpretation to a user's chain of actions over time by quantifying intentionality, thereby driving the interaction towards more meaningful concepts. Andy Deck's Open Studio (Giaccardi & Fischer, 2008) is another example of a mediator-based collaborative art project that uses affective mechanisms in a way that higher level interactions with users are interpreted on an emotional level which then reflects in the artistic content of the created artifact. Later, we show that our proposed collaborative interactive evolutionary framework of design is a mediator-based method of co-creation supported by affective mechanisms that connect various users in a larger conceptual space.

Collaborative Interactive Genetic Algorithms for Creative Design

Evolutionary algorithms such as genetic algorithms, genetic programming, evolutionary strategy etc. are a class of population-based optimization algorithms inspired by evolutionary biology that refine a set of potential solutions using representation, combination and selection operators based on relative fitness with respect to a problem. Creative evolutionary computational systems have been defined as evolutionary algorithms that either aid human creativity or solve design problems that only creative people could solve (Bentley, 1999; Bentley & Corne, 2002), a definition used to characterize creative computational systems in general. Goldberg presents an idealized framework for conceptual design in four components: problem, designer, alternative designs and design competition, and shows how evolutionary techniques, specifically genetic algorithms (GAs), can be thought of as "a lower bound on the performance of a designer that uses recombinative and selective processes" (Goldberg, 1991). Rosenman has explored evolutionary models for non-routine design (Rosenman, 1997a) and has investigated the generation of creative house plans (later referred to as floorplans in this paper) using genetic algorithms (Rosenman, 1997b). Creation of floorplans has also been investigated by Gero and Schnier as an evolving representation problem that restructures the search space (Gero & Schnier, 1995), by co-evolution of design and solution-spaces (Poon & Maher, 1997) and using case-based reasoning (De Silva Garza & Maher, 2000). A similar layout design problem – that of arranging furniture and objects around a room using genetic algorithms has been dealt in Nakajima et al. (2006). Genetic approaches are relevant to ET-creativity and are most likely to be described as models of creativity when applied to art, media and other domains that are normally regarded as involving human creativity.

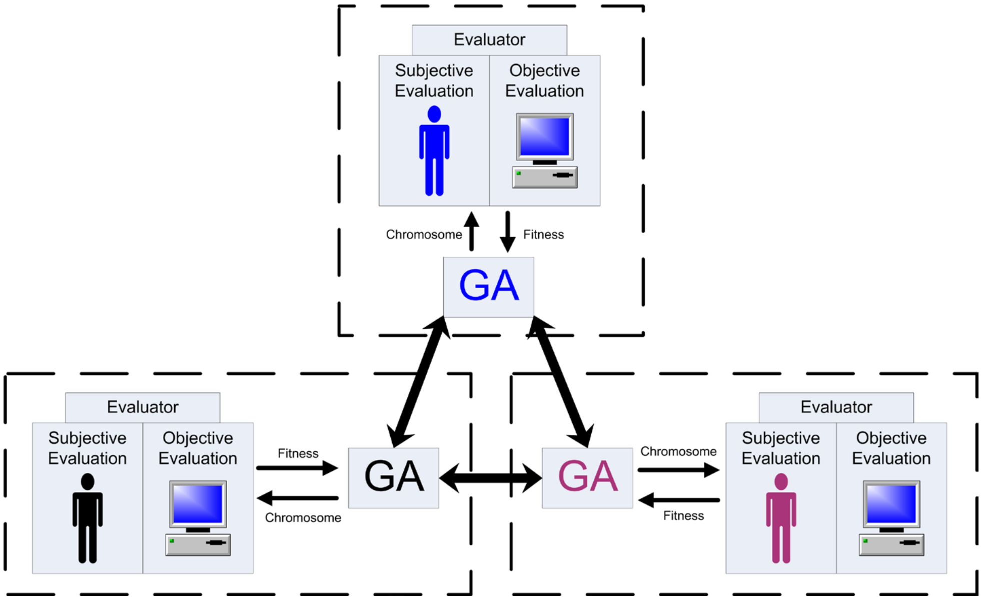
Unlike engineering design where optimization criteria are readily quantifiable, in domains such as architectural and product design various design concepts may need to be subjectively evaluated, especially when requirements include aesthetic and other subjective criteria. It is difficult, often impossible, to construct measures and explicit functions that can mimic the way designers evaluate subjective criteria. Interactive Genetic Algorithms (IGAs) are genetic algorithms whose fitness function is replaced by interactive user evaluations. IGAs in particular and interactive evolutionary computation (IEC) in general have been used in a wide range of applications ranging from engineering to arts and social sciences to design user-centric optimization systems (Takagi, 2001). Interaction modifies the GA in a fundamental way – by using subjective fitness, the fitness landscape is dynamic without a definite maximum. The system objective should therefore be to learn more about user preferences and constantly modify a loosely defined fitness function. Interactivity therefore relates directly to knowledge creation and knowledge consolidation in the fitness space.

A computational approach to investigate conceptual spaces in design (also called solution spaces in evolutionary terms) to support a human designers' exploration is presented by Woodbury and Burrow (2006). A conceptual space is defined as a networked structure of related descriptions of partial and intentional designs encountered in an exploration process. For example, a designer designing a new kind of chair considers every imaginable juxtaposition of objects that make sitting possible as a vast conceptual space of chair designs; he or she then starts with an idea or many ideas, continuously evolves them using partial concepts from within this conceptual space by traversing the space in some unstructured manner. We continue this line of thought with a collaborative approach to the exploration of solution spaces using an IGA-based system. The basic framework is shown in Figure 1. At the lowest level an interactive genetic algorithm searches a solution space with a continuously changing fitness landscape, belonging to a particular designer at use time. At a level higher, the collaborative tool that connects different designers together lets each redefine their solution space at use time, thereby achieving the equivalent to the meta-design proposition of user participation at system build-time. We attempt to explain why this process is creative according to Wiggins' formalized approach to creativity and also relate it to the meta-design open-system approach to co-creation as follows:

Let the initial conceptual space C_0^i for designer i be defined by set of rules $\tilde{R}_0^i$ and an interpretation function $\left|.\right|$, let space traversal be defined by set of rules $\hat{R}_0^i$ and the related interpretation function $\left\|.\right\|$; and let the quality assessment of potential solutions be defined by another set of rules $\overline{R}_0^i$.

1. $\overline{R}_o^i$ is a function of the fitness, which in the implementations described in this paper, comprises of a static knowledge-based objective component and a subjective component that changes with respect to time. This constantly evolving subjective part depends on how the designer interacts with the system. Assuming a "rational" user the interactions should be interpreted by the system as trying to reinforce the static component of the fitness function with unquantifiable subjective notions. Whether or not the knowledge base can be expanded based on these interactions is an open-research question. We therefore drop the subscript 0 and denote the quality assessment rules for designer i as $\overline{R}^i$. For example, in the collaborative support system developed for designing floorplans presented later in this paper, the objective component is derived from architectural guidelines and governmental regulations

Figure 1. Schematic showing implementation of the proposed collaborative interactive evolutionary model for creative design



regarding layout of rooms and open spaces, and the subjective component is built by interpreting the interactions as providing specific information about designer likes and dislikes regarding relative positioning of rooms and other interior details.

2. $\hat{R}_0^i$ and $\|.\|$ depend on genetic recombination operators such as crossover and mutation, and the selection criteria used to create the new population. In population-based searches, evolution of new concepts should be seen as evolution of new populations, the individuals in which will fare better with respect to rules $\overline{R}^i$ compared to individuals in previous populations. The traversal rules $\hat{R}_0^i$ can be modified to $\hat{R}_1^i$ by collaborating with a peer designer *j* searching a vastly different conceptual space C_0^j using a different set of traversal rules $\hat{R}_0^j$; mutual collaboration also lets *j* modify his or her set of traversal rules to $\hat{R}_1^j$. The designers can be remotely located, each using an interactive system that has been independently built as a closed system at built time. Collaboration allows the two remote systems to exchange algorithmic parameter information such as type of crossover operator, probabilities of crossover, selection probabilities, selection operator type etc., and then modify their own parameter information based on the new set of information made possible by collaboration. We hypothesize that although the independently developed interactive systems are closed systems, collaboration and exchanging such information at use time, eventually means that each individual system acts as an open system. In the floorplanning example, this can be seen as the interaction between a designer exploring similar looking variants of a two-bedroom, one open space plan with another designer exploring radically different bedroom plans, which is a result of selection and recombination parameters that can be biased towards either exploration or exploitation of the solution space. By exchanging parameter information, the

first designer can explore radical versions of the two-bedroom one-open space plan rather effectively.

3. The bounding rules for conceptual space $\tilde{R}_0^i$ and the interpretation $\left| . \right|$ depend on the genetic representation and the transformation function used to convert a plan in the genotypic space of solutions to a plan in the phenotype space (the conceptual space or the solution space). For designer *i* to modify the conceptual space from C_0^i to C_1^i thereby making T-creativity possible, the set of bounding rules can be changed to $\tilde{R}_1^i$ by modifying the genetic representation or the transformation function during use time. As can be envisaged, the latter is easier to implement than the former, and also forms the basis of creativity using association and analogy. For example, the same genetic representation of a floorplan can be transformed into the real-world as a plan with straight-walled rooms, or can be modified to be interpreted as plan with some rooms having at least one curved wall. Small empty spaces between rooms can be interpreted as closets or differently interpreted as ante-rooms. The interpretation depends on the transformation function, information about which can be exchanged at use time between remote peers.

Figure 1 represents a particular implementation of the proposed computational model of creative product design. Each dotted box represents a running instance of the interactive design tool – each instance is guided by a designer and searches a bounded conceptual subspace in accordance with designer preferences. The genetic algorithm in each instance combines the designers' subjective picks with a computable fitness function to drive the search through the conceptual subspace. The user-interface to our design tool allows the designer to zoom in on a particular displayed design and pick form-based aspects of the design for exploration. In addition to guiding his or her own search, each designer also can see a small subset of the other designers' evolving designs in (possibly) different conceptual spaces. The designer can then choose to modify his or her search space or move to another space by incorporating one or more of the peer-evolved designs into his or her genetic population. Incorporating peer-evolved designs also tends to influence the subjective and objective utility functions associated with the genetic search.

Algorithmic Implementations

In this section we briefly present results obtained using the proposed model and its implementation in two domains – designing two-dimensional architectural floorplans and editorial layout design for brochures. The non-dominated sorted multi-objective genetic algorithm, abbreviated NSGA-II (Deb, 2001) was used as the underlying genetic algorithm for the floorplanning problem. The NSGA-II creates Pareto-optimal fronts of non-dominated floorplans; within a front none of the plans are any worse than any other individual across all fitness criteria and all plans within a front are said to have the same rank. The multi-objectivity is a result of treating the objective criteria separate from subjective criteria. With regard to the objective criteria, the only measurable objective in floorplans is their compliance with architecture data guidelines (Neufert, Baiche, & Walliman, 2008). The guidelines for single-storey house plans relate to minimum room dimensions and areas. Every individual floorplan is assigned an objective fitness value based on its compliance with the minimum dimension and minimum area guidelines. With regard to subjective criteria, the algorithm lets the designer pick a particular floorplan as being the "best". This subjective pick (based on the user's preferences) is analyzed and translated into a user's preference for the number of rooms, total built area (area occupied by *rooms*), and room adjacencies. An individual plan in a population is compared to the "best" plan and assigned high subjective fitness values if the plan is similar to the "best" plan (selected by the user) in each of the three subjective criteria. Parents used to populate a new generation are selected by using the crowded distance tournament, a specialized genetic selector operator.

We define individual floorplans as recursive partitions of a 2-D rectangular panel using a binary tree representation, coded as a nested list. Tree representations have been used to evolve creative shapes of reading lamps (Liu, Tang, & Frazer, 2004; Liu & Liu, 2006). At the root node, the lamp is classified as being a combination of four components – a shade, a light, a holder and a bottom. Each of the four components is defined by features such as size, shape, color, among others. This rudimentary representation lends itself to recombination using generalized genetic programming operators. In the floorplanning implementation, the parameters at every node of the tree specify how the rectangular panel at that level is subdivided (either left/right or top/bottom) and what percentage of panel area at that level is contained in either the left or the top subdivision. Figure 2 shows how the rectangular panel is subdivided into *rooms* and *spaces*. A *room* is represented by the list [0, 1] and a *space* by [0, 0]. An arbitrary list [0, 0.75] represents division in top/bottom configuration with top sub-panel containing 75% of the parent panel. Another list [1, 0.80] represents division in left/right configuration with left sub-panel containing 80% of the parent panel.

The binary tree representation for floorplans necessitates the need for a specialized tree-crossover operator. The nested list is parsed as a binary tree and two such parent trees are crossed at randomly chosen nodes, such that entire sub-trees following those nodes are swapped. The tree representation is used in genetic programming (Koza, 1992) and hence, our crossover operator maps to the crossover operator used in genetic programming. The operator is shown schematically in Figure 3. Depending on the probability of mutation, the mutation operator works on the two parameters of the nodes (or leaves) differently. It performs a binary swap on the first parameter thereby changing the subdivision configuration. Depending on the value of the second parameter, the operator either performs a binary swap (if the value is either 0 or 1), thereby changing a *room* to a *space* and vice versa, or if the second parameter is a real number between 0 and 1, the operator replaces it by another random real number in the same interval, thereby altering the dimensions of the *room* (or the *space*).

Students, both graduate and undergraduate, at the Evolutionary Computational Systems Laboratory (ESCL) at the University of Nevada, Reno, were given instructions in using an interface for interaction and collaboration for the floorplanning problem. Student designers with apparently no knowledge of creating floorplans were asked to guide the system to design a floorplan for a two-bedroom, one-bathroom apartment with the following constraints: (1) one of the corners of the living area is also the north-west corner of the plan, (2) the two bedrooms should not be have a common wall, and (3) at least one of the bedrooms has a direct access to the bathroom. The problem stated above was solved both individually and collaboratively. During the collaborative evolution, only nine representative designs from the large population size maintained by the IGA were made visible to every designer in the peer-group. This was done to reduce user fatigue. However, during individual evolution of floorplans for the same problem (without collaboration), the designer had visual access to all the evolving designs in the population maintained by the user's IGA. The non-collaborative interactive and the collaborative interfaces are shown in Figures 4 and 5 respectively.

The representative visualization is done by selecting designs from the Pareto-optimal fronts created by the NSGA-II. Three designs are picked from the first front, three from the second front, and so on until nine designs are obtained. We also enforce that all individuals in the displayed subset are unique, given that the population contains enough diversity, since displaying a small subset consisting of numerous repeats is not useful to the user and does not give the user a sense of the current state of the population. By displaying a small subset and through fitness interpolation we can reduce

Figure 2. Binary tree representation of floorplans encoded as a nested list

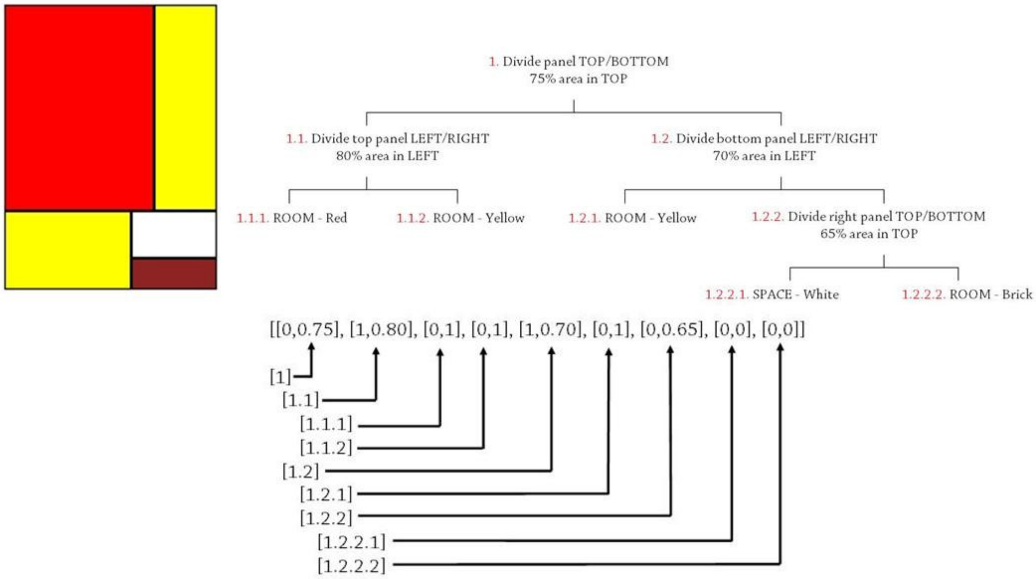


Figure 3. Tree crossover operator

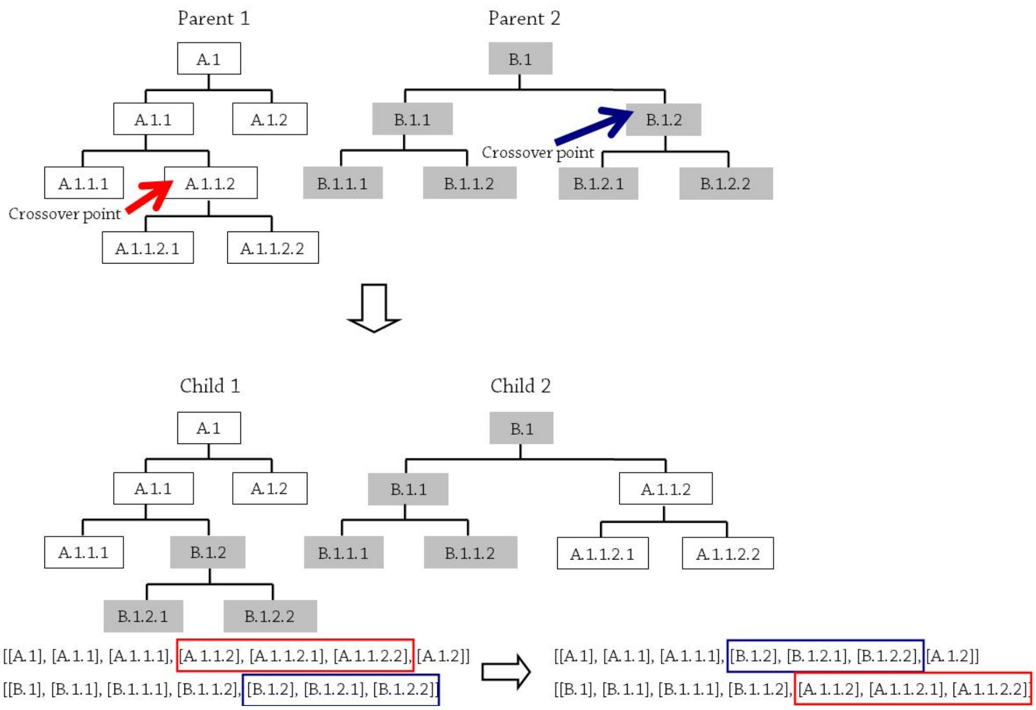


Figure 4. The non-collaborative interactive interface for floorplanning

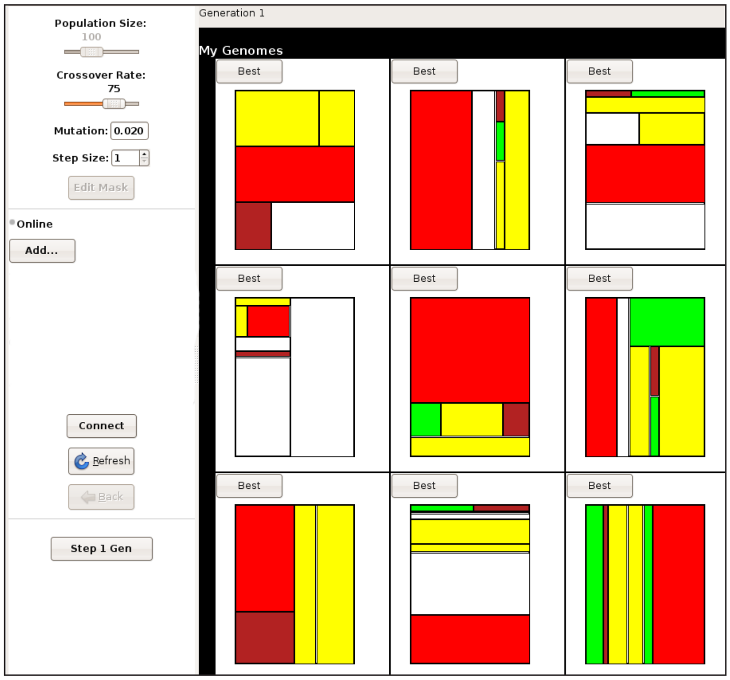
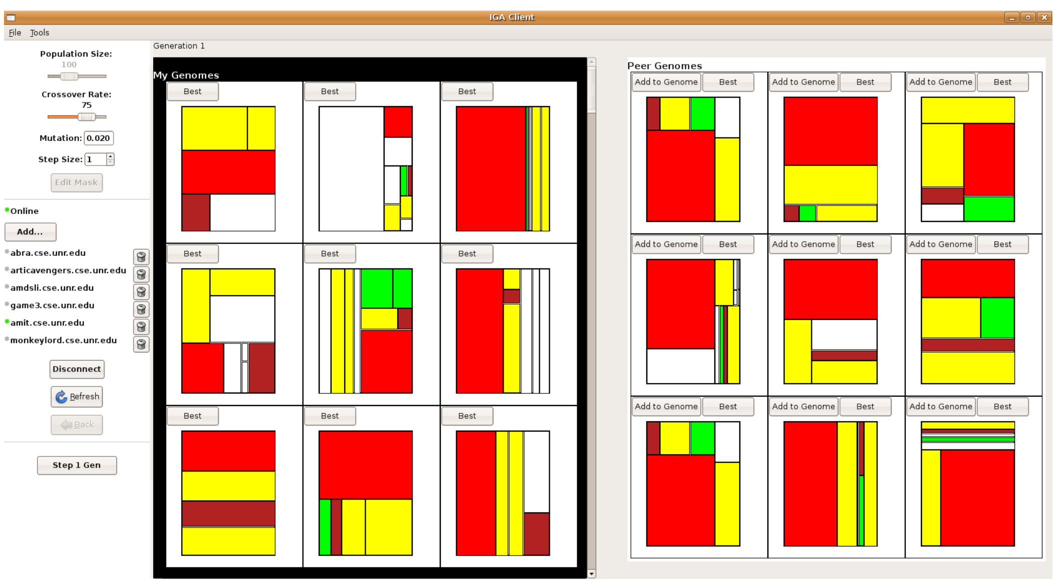


Figure 5. The collaborative interactive interface for the floorplanning problem



the amount of user interaction, and thereby, user fatigue. However, if the case arises that the user does not like any of the individuals in the displayed subset, then the user has the option to scroll down the current panel, and view the rest of the population, which remains hidden from view unless the user scrolls down. For users with little patience or that fatigue quickly (which is often the case), they can adhere to the use of the displayed subset. For the adventurous users, who are not intimidated by viewing hundreds of individuals to find the individual they like the best, they can scroll to view every single individual in the population. The ability to view the entire population also proves useful to users who early in a session explore the entire population, when there is a high degree of diversity, and later on only use the subset after the population has been biased to custom-evolved individuals.

The designer's interactive (non-collaborative) input consists of selecting the design he or she likes the best from either the subset of representative individuals, or from one of the individuals from the rest of the population (viewed by scrolling). In collaborative evolution, every designer is free to inject one or more of the designs visualized from the peer group into his or her evolving population. In addition, the designer also has the choice to select one of the case-injected peer designs as the floorplan considered the best by the user, which is useful in cases where the user's population has converged to undesirable solutions. The user selected best is then used to interpolate the fitness of every other design in the population. Through the interface we also support the ability to provide input every n^{th} generation, where n is the number of generations skipped before asking for user input, and which can be changed during a session. We also allow the user to go back to a previous generation if the population diverged into an undesired direction. The user also controls the crossover and mutation rate through the interface.

A fitness biasing scheme is employed to ensure that injected designs from peer populations survive long enough to leave a mark on the host population by using the concept of bloodline. Injected designs are considered to be full blood, while designs already in the population are treated as designs with *no blood*. The bloodline consists of a number between 0 (no blood) and 1 (full blood), and this value is added as another criteria to be maximized by the NSGA-II with Pareto optimality. Thus injected designs will all be non-dominated (in the topmost front) and will not die off immediately. The injected individuals replace the bottom 10% of the population (Louis & Miles, 2005). When a full-blooded individual crosses over with a no-blooded individual, then the offspring will inherit a bloodline value equal to a weighted sum of the bloodline of the parents, where the weight values depend on the percentage of the genetic material inherited from each parent.

The editorial design task for brochures and documents (posters) was implemented using a non-collaborative interactive genetic algorithm. Three genetic algorithms were used – the NSGA-II, a modified NSGA-II where each offspring design is created by mating (crossover followed by mutation) a design from the present population with the user-selected best, and an algorithm based on Dawkins's biomorph (Dawkins, 1996) where offspring designs are generated by applying pure mutations to the user selected best design. The three algorithms were selected to study the effect of generating offspring in different ways, which could have drastic effects in the behavior of the IGA, which can range from many diverse designs in the population to rapid convergence to a population that contains designs similar to the user selected best. Harrington et al. (2004) have proposed a set of aesthetic criteria for automated document layout which have been used as objective criteria. These include use of white space, the degree of overlap in the shapes (tiles) and spatial balance in the overall image. The same criteria are also judged subjectively by comparing the user selected best design to every other design in the population. The averaged objective and subjective criteria are optimized using Pareto-optimality.

By building on the framework developed for floorplanning, we were able to modify the representation to evolve document layouts for editorial design (Quiroz, Banerjee, Louis, & Dascalu, 2009). By taking the rectangular rooms created in a typical floorplan and allowing for various transformations to be applied to each room while drawing each room respectively, we were able to get some degree of overlap and interesting shape combinations. We also allowed for various shapes to be drawn, such as drawing a circle instead of a rectangle for where a room should be, in order to achieve an even greater degree of variation and possibly interesting tiles.

We allowed for three types of shapes: (1) rectangles, (2) ellipses, and (3) rounded rectangles. Each of these shapes can be scaled up or down along the x and/or y axis by up to 10%. The scaling allows for the original floorplan representation to be transformed into a collection of shape tiles, where either each tile can represent a placeholder for content (such as text or an image) or where the collection of tiles could represent a background design. The distinction between this representation and the floorplan representation is that the recursive partition, encoded as a nested list, is used only to create the initial set of shapes in a document. Once we know the allocation of tile (rooms in floorplans), we assign the shapes to one of four quadrants, based on the shapes' locations, using the shape's center as the point of reference. This results on a representation based on quad-trees of depth one. This is shown in Figure 6.

The editorial design interface had an additional feature – the designer was able to customize evolved designs to make them look like actual brochures or posters. This functionality was lacking in the floorplan implementation which made the implementation less realistically satisfying to the designers. Evolved brochures or posters can be edited for content by the designer in a larger screen, which helps him or her to appreciate detail and facilitate editing. The background tiles can be moved around, text can be added to one of the existing shapes, images can be overlaid on tiles, tiles can be resized, deleted or have their colors changed among others. After customizing the designer has the option to save the current document as an image. Some examples of brochures created by students at ECSL are shown in Figure 7. The collaborative framework can easily be extended to the brochure design task, or even better the two tasks can be combined because they use essentially the same genetic representation that is interpreted differently. Collaborating between the two tasks might lead to evolution of floorplans with rooms that have curved walls and organic shapes.

The proposed model and its implementation raise several issues related to interaction, collaboration, creativity, knowledge creation and computational intelligence. These are addressed below:

Figure 6. Depth one quad-tree representation for document layout design

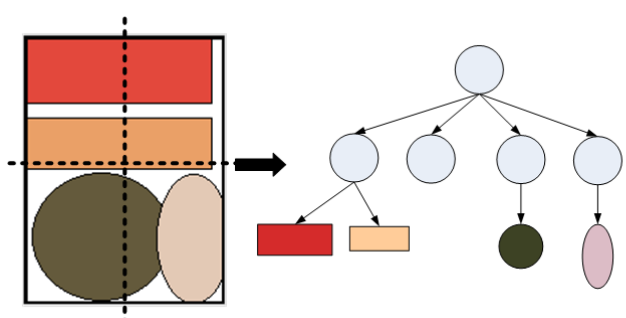
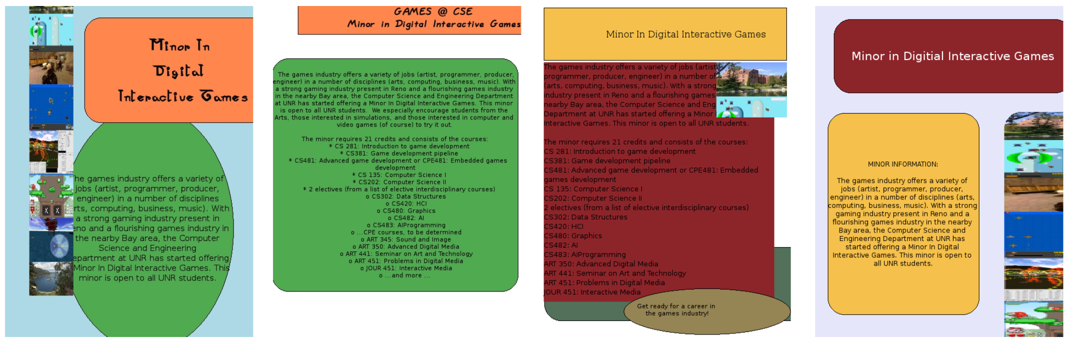


Figure 7. Sample brochures evolved and customized using the interactive genetic algorithm for editorial design



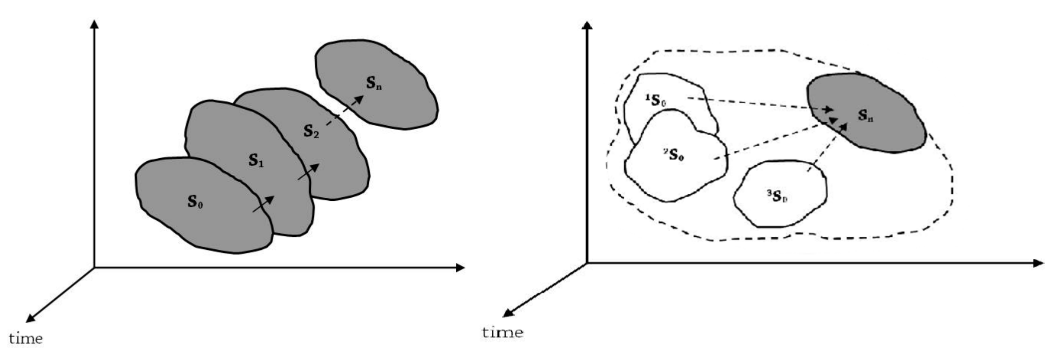
User Fatigue: Interactive computer systems need explicit mechanisms to sustain user interest throughout use time. Since the program performs most of the basic design tasks and only involves the user in the design evaluation phase which is a repetitive task, it has been seen that users (i.e. human designers) tend to lose interest as the task progresses. There has been a whole body of research that deals with mitigating user fatigue in IGAs and other interactive computer-supported design systems (Gu & Frazer, 2006; Watanabe, Yoshikawa, & Furuhashi, 2007). Probabilistic and statistical methods combined with machine learning techniques have been used in the past; such techniques usually built using neural networks or support vector machines try to infer preference rules from a limited subset of interactions with the user. The objective is therefore to limit interaction instead of promoting it. The designer becomes less and less central to the process with time – the evaluation program learns from his or her few interactions, thereby building a virtual model of the user and predicting his or her actions as the design process progresses. The other approach is to make interaction less frequent but keep it constant throughout the process. The designer is only shown a small representational subset of the entire population at any given time and asked only to choose a concept that he or she "likes the most." In addition designers may also be asked to choose another concept that they "dislike the most." The intention is to learn just enough about some predefined criteria in the subjective component so as to allow the algorithm to rank solutions in the entire population by interpolation. Two related issues are – how to select a representative subset that encapsulates information carried by the entire population; and how to best define a distance measure to be used to rank designs based on the limited interaction. Unsupervised classification (clustering) techniques have been used in the past to address the former issue. Clustering techniques such as k-means produce k-distinct clusters in the population – solutions within a cluster are similar to each other and dissimilar to solutions in other clusters. The representative solutions from each of the clusters then are used to be part of the much smaller interactive subset. A suitable distance measure can be defined in the genotype space, using parameters that are part of the genotypic representation. For example, longest common subsequence (LCS) distance and Hamming distance can be used measure proximities between a pair of concepts whose genotype is made up of a linear sequence of integers (real or binary). Distance measures can also be in theory defined in the phenotype (solution) space.

Subjective Fitness Evaluation: Evaluation of alternative concepts is central to any design process. Even ill-defined design problems such as in domains explored in this papers, have an objective body of knowledge related to the design process which help create general guidelines for the process to progress. For example, for the floorplanning problem, certain basic architectural guidelines need to be followed whereas in editorial design, rules relating to color juxtaposition have been well-established and need to be incorporated in the objective fitness component. However, in addition to such objective guidelines, designers use their domain expertise and unquantifiable knowledge, preferences, emotions and biases to root out "bad designs" from the "good ones." The implementation issues are two-fold: how to combine subjective preferences with objective guidelines, and how to interpret user preferences about aesthetics and other unquantifiable elements just by looking at user-preferred designs. For example, a designer might chose a floorplan over another given a choice; how does the system interpret this choice – is he or she choosing a particular floorplan because of the way the rooms the laid out with respect to each other, or because of the fact that the living room is easily accessible from all three bedrooms or something even more subtle. The designers themselves do not make usually make a conscious choice; there is quite possibly interplay of a variety of factors that lead them to choose a certain floorplan over other. This leads us to envision a system that can learn as the design processes progresses, track user-preferences over time (usually a very short use-time) and make intelligent surmises. Another approach is to start with a few very low-level guidelines and then constantly modify them as new information about user-preferences start coming in. This is akin to domain knowledge creation facilitated by expert-users. The system can also be built to evolve knowledge using expert-users in a supervisory role. In research presented in this paper, we however treat the body of knowledge used to create objective rules as sacrosanct and assume that user-preferences about design elements do not include guidelines that are the basis of the objective component of fitness. In other words, the objective component is distinct from the subjective one. We then proceed to combine rules in the form of metrics, from the two components. There are two possible ways to do this – one is to create a weighted linear combination of such metrics from the two subdomains, but the issue is how to assign weights to signify relative importance of the different criteria? Another approach is to treat the two components separately and analyze Pareto-optimal fronts produced by the set of metrics such as in multi-objective optimization. We have investigated both approaches; however, there is no conclusive evidence for one approach being better than the other, and hence remains an open research question.

Collaboration and ET-Creativity: We have already presented arguments that relate Wiggins's formalized framework of creativity with the collaborative interactive genetic search-based model of creativity proposed in this paper. Here we look at some research issues concerning implementations of the proposed collaborative model. The implementation of the model is at least pseudo T-creative; the remotely located designers working on the same design problem (floorplan design or editorial design) have the ability to view a set of evolving peer designs and include one or more of these concurrent designs in their own population at any given time. In the present implementation that does not change system parameters or add extra variables to the design space itself. It does provide a departure (possibly radically) from the established subjective fitness norm. This leads to changing the set of traversal rules which Wiggins argues is E-creativity, although it can be significant in the sense that it could guide the search to new areas of the already defined conceptual space. We argue that although the conceptual space of floorplans for a particular designer is the same as the space for another designer, the real "usable" subspace of designs is much smaller. We call this usable conceptual space, the design space and changing the set of traversal rules changes the design space. T-creativity as a non-collaborative process is shown schematically in Figure 8 (left). The collaborative modification of design space is shown in Figure 8 (right) with designer 1 moving from an initial design space, ${}^{1}S_0$ to S_n by collaborating with two other designers working in the same conceptual space, both of whom also converge to the design space S_n. True T-creativity can be achieved in two possible ways within our collaborative interactive evolutionary exploration framework, (1) different designers start with different underlying representations, and representation of the designer who injects peer-designs is modified drastically so that he or she can now search a completely new (previously unknown) conceptual space, or (2) the underlying representation is the same across the network, but genetic parameters are either switched off or on and injecting peer-designs will switch on (and off) a different set of design parameters, thereby extending or moving the conceptual space. The latter can be implemented by an efficient masking scheme in the representation. Collaboration with interactive evolutionary search has the potential to be a T-creative design process.

P and H-creativity: A genetic search on a predefined conceptual space from a set of user-specified concepts is enough to guarantee P-creativity if it can be argued that the evolved concepts are both novel and valuable. Content novelty and value in turn can be guaranteed in theory if the user interacts with the search. On the other hand, H-creativity measured in the socio-historical sense is also theoretically possible if collaboration between two (or more) users working in different domains leads to the evolution of novel and valuable concepts that have never been proposed before in their respective domains. Interaction with a genetic search makes a P-creative scenario possible, and collaboration among users might lead to H-creativity if the collaborative system supports such cross-domain interpretation of concept-representation transformations. In fact, H-creativity as a result of cross-domain collaboration is a special case of combinational creativity, as seen in concepts which are born out of association and analogy.

Figure 8. Left: T-Creativity in design processes involves changing state spaces of possible designs i.e. design space with time; Right: Collaborative creative exploration as a pseudo T-creative process



Mediators and Affective mechanisms: Designers collaborating with other designers, interacting with a guided genetic search is a co-creation setup, where each designer is engaged in a meaningful activity while being influenced both by the interaction and the collaboration. The emergence of new designs in a particular designer's evolving population is a function of both the interaction and collaboration. The design population should necessarily correspond to the personal state-of-mind of the designer at any given time. A designer might collaborate for some time before switching off collaboration due to changes in his or her emotional or rational state. ("I do not want to collaborate with designers who have no idea what they are doing" or "my designs are quite superior to what my peers are producing so why bother collaborating.") As long as the interactive choices are rational and can be interpreted by the IGA, collaboration or non-collaboration will reflect in the choices presented to the designer in the next round of design visualization. Classes of mediators in co-creation are defined based on spatial and chromatic relationships, and figurative, textual and temporal elements (Fischer & Giaccardi, 2006) – which can all be envisioned to be present in a creative open-ended activity that involves collaboration with other designers and interacting with the computer-based support system. Open-ended design activities are defined as activities that have a defined purpose and no explicit goals, activities that require imagination and creativity.

Knowledge Creation: The collaborative interactive genetic search process proposed in this paper can also be implemented in an online wiki-format (Watson & Harper, 2008). This is infinitely more beneficial for difficult-to-solve constrained optimization problems that rely on distributed knowledge and user input. By collaborating across a common platform, users consolidate and create a new body of knowledge that can be subsequently applied to related problems in different domains. It can also provide different, remotely located communities of practice a situated forum to bring to the table their diverse knowledge tools for collaboratively completing complex tasks. As an example, imagine instructors of sociology, engineering, arts and humanities from different universities across the world collaborating to develop effective teaching strategies using case studies. The case studies from the different domains can be manipulated by an IGA by selecting, recombining and mating aspects deemed interesting in the interaction between instructors and the IGA. The result of the evolution say after 10 generations is a vast body of knowledge that is more than the sum of its parts.

CONCLUSION

In this paper, we propose a collaborative model for creative design based on interactive genetic algorithms. The collaborative computational model has been shown to possess exploratory-transformation creativity, and is also related to the meta-design approach for solving complex design tasks using open support systems. We have addressed several issues relating to implementation such as ones relating to creative spaces, user fatigue, subjective fitness evaluation, mediators, affective mechanisms, interaction and collaboration. We implement our proposed model to collaboratively evolve floorplans and interactively evolve editorial design layouts for brochures and posters – two applications that use an established body of knowledge and rely on user preferences. Although these implementations are designed at the product level (a floorplan or a brochure is a product), the same principles can be used to design for at an abstracted level, e.g. instead of directly evolving floorplans as physical artifacts, they can be evolved at the "idea" level in terms of their constitutive elements. This is almost certainly more realistic and also mimics the creative process architectural teams go through at design conceptualization time. The amount of creative content is more at the abstracted level compared to the physical level of design, involves more imagination than knowledge. In summary, an open system is simulated by implementing a synthesized (closed) system used collaboratively across different knowledge domains and across users with different subjective preferences to the same design process. An open computer support system is central to the premise of meta-design, with the promise of enhanced knowledge creation and computational system intelligence.

ACKNOWLEDGMENT

This work was supported in part by contract number N00014-0301-0104 from the Office of Naval Research and the National Science Foundation under Grant no. 0447416.

REFERENCES

Arthur, W. B. (1994). On the evolution of complexity. In Cowan, G., Pines, D., & Meltzer, D. (Eds.), *Complexity: Metaphors, models, and reality* (pp. 65–81). Cambridge, MA: Perseus Books.

Banerjee, A., Quiroz, J. C., & Louis, S. J. (2008). A model of creative design using collaborative interactive genetic algorithms. *Design Computing and Cognition*, *5*, 397–416. doi:10.1007/978-1-4020-8728-8_21

Bentley, P. (1999). *Evolutionary design by computers*. San Francisco, CA: Morgan Kaufmann.

Bentley, P., & Corne, D. (2002). An introduction to creative evolutionary systems. In Bentley, P. J., & Corne, D. W. (Eds.), *Creative evolutionary systems* (pp. 1–76). New York, NY: Academic Press. doi:10.1016/B978-155860673-9/50035-5

Boden, M. A. (1999). Computer models of creativity. In Sternberg, R. J. (Ed.), *Handbook of creativity* (pp. 351–372). Cambridge, UK: Cambridge University Press.

Csikszentmihalyi, M. (1997). *Creativity: Flow and the psychology of discovery and invention*. New York, NY: Harper Perennial.

Dawkins, R. (1996). *The blind watchmaker: Why the evidence of evolution reveals a universe without design*. New York, NY: W.W. Norton & Company.

De Silva Garza, A. G., & Maher, M. L. (2000). Characterising evolutionary design case adaption . In *Artificial intelligence in design*. Dordrecht, The Netherlands: Kluwer Academic.

Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*. Chichester, UK: John Wiley & Sons.

Fischer, G., & Giaccardi, E. (2006). Meta-design: A framework for the future of end-user development. In Lieberman, H., Paternò, F., & Wulf, V. (Eds.), *End user development: Empowering people to flexibly employ advanced information and communication technology* (pp. 421–452). Dordrecht, The Netherlands: Kluwer Academic.

Gero, J. S., & Schnier, T. (1995). Evolving representations of design cases and their use in creative design. In *Proceedings of the 3rd International Conference on Computer Models of Creative Design* (pp. 343-368).

Giaccardi, E., & Fischer, G. (2008). Creativity and evolution: A metadesign perspective. *Digital Creativity*, *19*(1), 19–31. doi:10.1080/14626260701847456

Goldberg, D. E. (1991). Genetic algorithms as a computational theory of conceptual design. In *Proceedings of the 6th International Conference on Artificial Intelligence in Engineering* (pp. 3-16).

Gu, Z., Xi Tang, M., & Frazer, J. H. (2006). Capturing aesthetic intention during interactive evolution. *Computer Aided Design*, *38*(3), 224–237. doi:10.1016/j.cad.2005.10.008

Harrington, S. J., Naveda, J. F., Jones, R. P., Roetling, P., & Thakkar, N. (2004). Aesthetic measures for automated document layout. In *Proceedings of the ACM Symposium on Document Engineering* (pp. 109-111).

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

Liu, H., & Liu, X. (2006). Generative design supported by tree based genetic algorithm. In *Proceedings of the 6th IEEE International Conference on Computer and Information Technology* (p. 267).

Liu, H., Tang, M., & Frazer, J. H. (2004). Supporting creative design in a visual evolutionary computing environment. *Advances in Engineering Software*, *35*(5), 261–271. doi:10.1016/j.advengsoft.2004.03.006

Louis, S. J., & Miles, C. (2005). Playing to learn: Case-injected genetic algorithms for learning to play computer games. *IEEE Transactions on Evolutionary Computation*, *9*(6), 669–681. doi:10.1109/TEVC.2005.856209

Nakajima, T., Hashimoto, S., Haruyama, K., Nakamura, T., & Osana, Y. (2006). Office layout support system using interactive genetic algorithm. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 56-63).

Neufert, E., Baiche, B., & Walliman, N. (2008). *Architects' data*. New York, NY: Wiley-Blackwell.

Peng, C. (2001). *Design through digital interaction: Computing communications and collaboration on design*. Portland, OR: Intellect Books.

Poon, J., & Maher, M. L. (1997). Co-evolution and emergence in design. *Artificial Intelligence in Engineering*, *11*(3), 319–327. doi:10.1016/S0954-1810(96)00047-7

Quiroz, J., Banerjee, A., Louis, S., & Dascalu, S. (2009). Document design with interactive evolution . In Damiani, E., Jeong, J., & Jain, L. C. (Eds.), *New directions in intelligent interactive multimedia systems and services* (pp. 309–319). New York, NY: Springer. doi:10.1007/978-3-642-02937-0_28

Rosenman, M. A. (1997a). An exploration into evolutionary models for non-routine design. *Artificial Intelligence in Engineering*, *11*(3), 287–293. doi:10.1016/S0954-1810(96)00046-5

Rosenman, M. A. (1997b). The generation of form using an evolutionary approach . In Dasgupta, D., & Michalewicz, Z. (Eds.), *Evolutionary algorithms in engineering applications* (pp. 69–85). Berlin, Germany: Springer-Verlag.

Sims, K. (1991). Artificial evolution for computer graphics. *Computer Graphics*, *25*(4), 319–328. doi:10.1145/127719.122752

Takagi, H. (2001). Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation. *Proceedings of the IEEE*, *89*(9), 1275–1296. doi:10.1109/5.949485

Taylor, T., Soddu, C., Whitelaw, M., & Gabora, L. (2002). Creativity in evolution: Individuals, interactions, and environments. In Bentley, P. J., & Corne, D. W. (Eds.), *Creative evolutionary systems* (pp. 79–108). New York, NY: Academic Press. doi:10.1016/B978-155860673-9/50037-9

Todd, S., & Latham, W. (1994). *Evolutionary art and computers*. Orlando, FL: Academic Press.

Watanabe, Y., Yoshikawa, T., & Furuhashi, T. (2007). A study on application of fitness inference method to PC-IGA. In *Proceedings of the IEEE Congress on Evolutionary Computation* (pp. 1450-1455).

Watson, K., & Harper, C. (2008). *Supporting knowledge creation: Using wikis for group collaboration (research bulletin)*. Boulder, CO: Educause Center for Applied Research.

Wiggins, G. A. (2001). Towards a more precise characterisation of creativity in AI. *Case-Based Reasoning: Papers from the Workshop Programme at ICCBR, 1*, 113-120.

Wilson, P. (1991). *Computer supported cooperative work: An introduction*. Dordrecht, The Netherlands: Kluwer Academic.

Woodbury, R. F., & Burrow, A. L. (2006). Whither design space? *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, *20*(2), 63–82. doi:10.1017/S0890060406060057

Amit Banerjee is an assistant professor of Mechanical Engineering at the Pennsylvania State University at Harrisburg. His research interests include pattern recognition, computer-aided design, machine learning and interdisciplinary research in such diverse fields as computational chemistry, civil and structural engineering and bioengineering. He was previously with the Evolutionary Computing Systems Laboratory in the Department of Computer Science at the University of Nevada, Reno as a postdoctoral fellow where he conducted research on collaborative creativity and creative design systems. He serves as the technical chair of the international joint conference on advances in engineering and technology since 2010 and the international joint journal conference in engineering, an e-conference, since 2009.

Juan C. Quiroz is an Intellectual Property Specialist at SilverSky Group LLC in Reno, Nevada. His research interests include evolutionary computation, evolutionary art, computer-aided design, and machine learning. He has a Ph.D. in Computer Science and Engineering from the University of Nevada, Reno. He is registered to practice before the United States Patent and Trademark Office, and works on patent preparation and patent prosecution of inventions related to software, hardware, networks, communications, mechanical devices and medical devices.

Sushil J. Louis is a full professor of Computer Science at the University of Nevada, Reno and director of the evolutionary computing systems laboratory. He works in the areas of evolutionary computation and its applications in search, optimization, and machine learning. His current interests are in evolutionary approaches to computer game AI and human behavior modeling. Major application areas include computer game AI and human computer interfaces. Dr. Louis is an associate editor of the IEEE transactions on Computational Intelligence *and* Artificial Intelligence in Games *and was co-general chair of the 2006 IEEE Symposium on Computational Intelligence and Games. You can visit his website, http://www.cs.unr.edu/~sushil/ for more.*