A model of creative design using collaborative interactive genetic algorithms

Amit Banerjee, Juan C. Quiroz and Sushil J. Louis
University of Nevada, Reno, USA

We propose a computational model for creative design based on collaborative interactive genetic algorithms, and present an implementation for evolving creative floorplans and widget layout/colors for individual UI panels. We map our model and its implementation to earlier models of creative design from literature. We also address critical research issues with respect to the model and its implementation – issues relating to creative design spaces, design space exploration, design representation, design evaluation (competition), design collaboration, and design visualization (for interactivity). Results comparing collaborative evolution of floorplans to non-collaborative evolution are also presented, and pre-tests using surveys indicate that floorplans developed via collaboration are more *original* than those produced by individual non-collaborative evolution.

Introduction

Design is a fundamental, purposeful, pervasive and ubiquitous activity and can be defined as the process of creating new structures characterized by new parameters, aimed at satisfying predefined technical requirements. It consists of several phases, which differ in details such as the depth of design, kind of input data, design strategy, procedures, methodology and results [1]. Usually the first stage of any design process is the preliminary or the conceptual design phase, followed by detailed design, evaluation and iterative redesign [2]. Computers have been used extensively for all these stages of design except the creative conceptual design phase. According to Goldberg [3], this phase of design has been regarded as a *black art locked up in a time warp of platitudes, vague design procedures and problem-specific design rules*.

Design Computing and Cognition DCC'08. J.S Gero (ed),
pp. xx-yy.

Creative evolutionary computational systems have been defined by Bentley and Corne, as evolutionary systems that either aid human creativity or solve problems that only creative people could solve [4]. 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) can be thought of as 'a lower bound on the performance of a designer that uses recombinative and selective processes' [3]. Rosenman has explored evolutionary models for non-routine design [5] and has investigated the generation of creative house plans (later referred to as floorplans in this paper) using genetic algorithms [6]. Creation of floorplans has also been investigated by Gero and Schnier as an evolving representation problem that restructures the search space in [7], by co-evolution of design and solution-spaces in [8], and using case-based reasoning by De Silva Garza and Maher in [9].

Unlike detailed design where optimization criteria are readily quantifiable, alternative design concepts during the preliminary design phase may need to be subjectively evaluated, especially when requirements include aesthetic and other subjective criteria. It is difficult, often impossible to construct matrices 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 [10].

At the same time, collaborative systems have been the focus of studies into creativity and computer supported cooperative work [11] since the early 90s. There has been a paradigm shift from computer-aided design systems to computer supported collaborative design systems [12]. It has been argued that much of our intelligence and creativity results from interaction and collaboration with other individuals [13]. In this paper, we propose a computational model of the creative design process using a collaborative interactive human-centered approach to exploration of design spaces. We present a collaborative interactive genetic algorithm implementation for our model to evolve floorplans and widget layout/style design, as a user-interface development tool. We compare designs evolved by a collaborative peer group, against those evolved individually by designers, and find that the former designs consistently rated higher on the "originality" scale, thereby lending credence to our computational model.

Motivation

The purpose of the research presented in this paper is to build a collaborative, interactive, genetic algorithm based design tool to test the hypothesis that collaborative, interactive, evolutionary exploration of design space is a viable computational model of creative design. One of the distinctions made between different types of creativity include Boden's [14] two types of creativity: *H*-creativity and *P*-creativity. *H*-creativity or historic creativity occurs when the design falls outside the range of designs created by anyone in the society, whereas *P*-creativity or personal creativity occurs when the design is novel to the designer (but may not be novel to the world). *S*-creativity or situated creativity [15], a more recently identified type, occurs when the design contains ideas that were not expected to be in the design when the design was commenced. Thus the design may not be novel in the *P* or the *H* sense but is novel in that particular design situation. In the absence of cohesive collaboration, artists or creative people exhibit *P*-creativity. We are interested in investigating the social aspects of creativity by facilitating and encouraging group interaction and cooperation, which we hypothesize, will lead to individual *P*-creativity and a group *S*-creativity. Our model maps to a system of creative design through social acts and social influence [13,16], where individual designers interact with an evolutionary system to guide the *P*-creative design process while at the same time cooperating within themselves by introducing new state variables, thereby guiding the *S*-creative design process.

Our collaborative interactive evolutionary exploration model also relates to the *blind variation and selection retention* model based on the Darwinian theory of creativity [17]. The *blind variation and selection retention* model of creativity states that the creative process is characterized in the first stage by production of a myriad of ideas and thoughts while lacking the foresight in the production of variations, followed by subjective selection and retention of the most meaningful ideas and thoughts. Since 1960, a body of research has been dedicated to furthering the evolutionary model of creativity [18]. This Darwinian framework for modeling creativity has found use in a connectionist approach to create a computer-based model of the creative process [19] and connects well with our model. In the next section we present the discussion on the proposed computational model and issues related to its implementation.

The Computational Model of Creativity

A computational approach to investigate design spaces (or solution spaces) to support a human designer's exploration is presented in Woodbury and Burrow [20]. A design space is defined as a networked structure of related descriptions of partial and intentional designs encountered in an exploration process. Woodbury and Burrow also claim that robust reuse of paths of exploration is of critical importance in design space exploration. We continue this line of thought with a collaborative approach to the exploration of a solution space using an interactive genetic algorithm. The designer-centric aspect due to the interactivity with the genetic algorithm helps in assigning a utility to a particular solution in the space, while collaboration between various designers helps the genetic algorithm to diversify and explore an extended search space. We hypothesize that the collaboration brings about concurrent exploration of design subspaces, and elements of the final design are a creative (unforeseen) collection of individual elements of various subspaces.

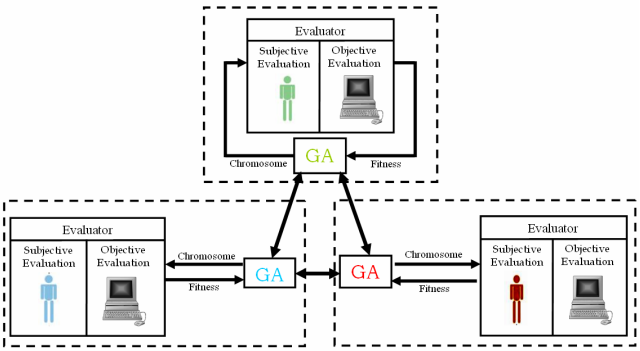

Figure 1. Schematic showing implementation of the proposed collaborative interactive evolutionary model for creative design

Figure 1 represents an implementation of our computational model of creative design. Each dotted box represents a running instance of the interactive design tool – each instance is guided by a designer and searches a particular subspace in accordance with designer preferences. The genetic algorithm in each instance combines the designer's subjective picks with a computable fitness function to drive genetic search through a design subspace. The user-interface to our design tool allows the designer to zoom in on a particular displayed design and pick aspects of the design for exploration. In addition to guiding his or her own search through a subspace, each

designer also can see a small subset of the other designers' evolving designs. The designer can then choose to expand 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 tends to influence the subjective and objective utility functions associated with the genetic search. The proposed model and its implementation raise several research issues. We divide these issues into five broad categories.

Design Space Exploration

The most important research issue is the collaborative versus individual exploration of the design space. Having investigated both collaborative and individual interactive genetic algorithms for evolving floorplans and widget layouts/styles for UI, we have empirical evidence that the creative content of the collaboratively evolved designs are superior to those of the individually evolved designs. We present evidence of our claim in the results section later in this paper.

Creative Design Spaces

According to Gero [21], creative designing can be defined in computational terms as the activity that occurs when one or more new variables are introduced into the design. This leads to the distinction between product and process creativity – creative design processes (processes based on addition and deletion of design variables) have the potential to aid in the design of creative artifacts, but as such they do not guarantee that the artifact produced is creative by itself. In other words, the creative design process is characterized by an extension or movement of the state space of potential design to new regions in the infinitely large state space of all possible designs. This is shown as a pictorial in Figure 2 (left).

Collaboration provides an effective framework for extending an existing state space or moving to a new state space. By injecting peer-designs into his or her genetic population, the designer is in a way modifying his or her own "effective" state space. This is shown as a schematic in Figure 2 (right) with designer 1 moving from his initial state, ${}^{1}S_0$ to S_n by collaborating with two other designers, both of whom also converge to the creative space. This feature can be implemented in two possible ways within our collaborative interactive evolutionary exploration framework: (1) different designers have different underlying representation schemes to start with, and the representation of the designer who is injecting peer-designs is influenced drastically so that he or she can now search a previously unknown solution space; or (2) the underlying representation is the same

across all users, but design parameters are either switched off or on and injecting peer-designs will switch off (and on) a different set of design parameters, thereby extending or moving the state space of solutions. This can be implemented by an efficient masking scheme in the representation. In summary, collaboration in evolutionary search has the potential to be a creative designing process.

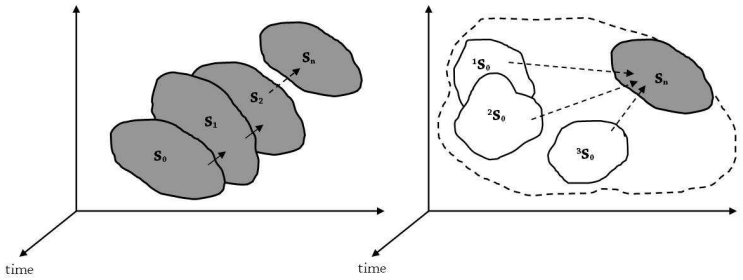

Figure 2. Left: Creative designing involves changing state spaces of possible designs with time [21]; Right: Collaborative creative exploration

Fitness Function

Evaluation of alternative designs is central to the conceptual design process. In addition to objective guidelines (which usually are derived from design requirements analysis), designers use their domain expertise and knowledge, preferences, emotions and biases to root out bad designs from the good ones. We incorporate evaluation based on subjective biases by letting the designer interact with the evolving population at either every generation of the evolution process or after every n generations ($n > 1$). However, since not every design criterion is subjective, we are faced with the problem of either combining the metrics obtained from the subjective and objective evaluation criteria or treating them separately. In the former case, how does one create a weighted linear combination of such metrics, and decide how to assign weights to signify relative importance of the different criteria? In the latter case, one might be tempted to treat the criteria separately by analyzing Pareto-optimal fronts (if any) produced by the set of criteria under consideration. 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.

Representation

The issue of what constitutes a good representation is vital to the efficacy of any evolutionary search technique. Evolutionary optimization techniques, including genetic algorithms, require that designs (or solutions) be encoded in a manner suitable for genetic operators, such as crossover and mutation, to work on. For the floorplanning problem, we have used a binary tree with much initial success as the genotypic representation that encode for the floorplans. We have also used the integer and bit-string representations to evolve widget layout/style.

In the next section, we present the collaborative interactive genetic algorithm implementation of our proposed model, which we call IGAP – Interactive Genetic Algorithm Peer to Peer.

IGAP: Interactive Genetic Algorithm Peer to Peer

We present an implementation of our collaborative model for creative design for evolving floorplans and widget layout/style schemes. Although, we have implemented them as two distinct modules, they can be seamlessly put together as a coherent two-phase user-interface design tool. In addition to being used as an archetype for UI panel layout, floorplanning is of importance to Architecture and Civil Engineering. The implementation is shown in Figure 3. We first present details of the IGA framework and discuss the collaborative framework later.

IGA Framework

IGAP is part of a GA/IGA framework we have built to support evolutionary design of user interface elements. For a problem requiring interaction with a user, the designer is required to implement the fitness function - which takes the user input and evaluates each individual in the population based on the user provided feedback, and a drawing function - which draws to the screen the subset of individuals from the population to be evaluated by the user.

Representation

For evolving floorplans we have used a binary tree representation, coded as a nested list. At every node of the tree, the parameters 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 4 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.

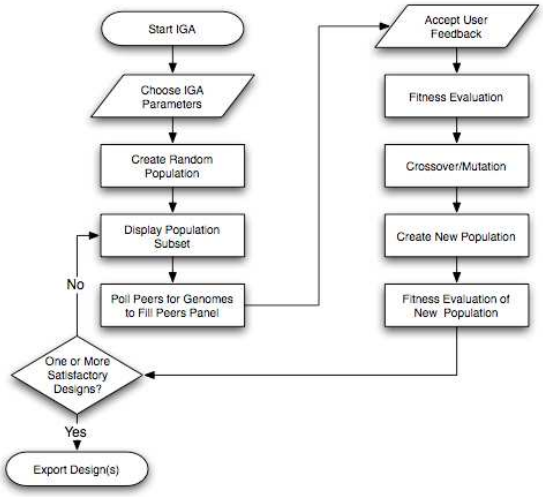

Figure 3. The collaborative interactive genetic algorithm implementation for creative designing

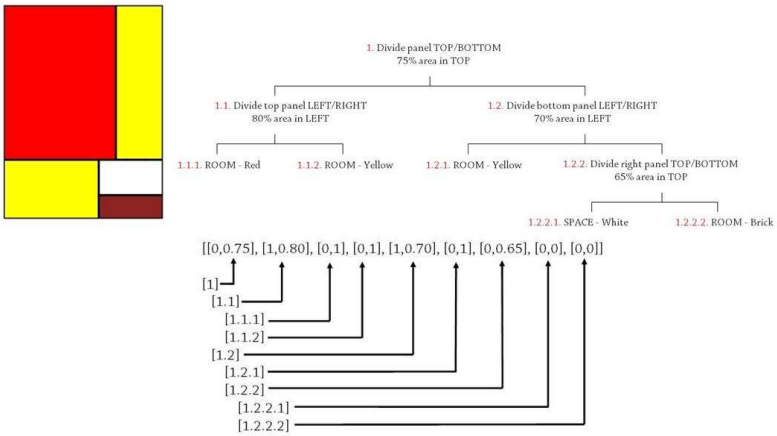

Figure 4. Binary tree representation of floorplans encoded as a nested list

For widget layout and style design, we use two chromosomes to specify the user-interface panel. On the panel, the widgets are laid out on a grid and different widgets are identified by a widget identification number (1 onwards, 0 for spaces on the panel grid). A sample layout and its encoding are shown in Figure 5. This is an example of an integer representation. The second chromosome encodes for various style characteristics, including background and foreground color, vertical and horizontal spacing between widgets in the layout grid, and font type. All of these attributes are encoded in a bit string – string of 0s and 1s. For color we use the RGB representation, where each color consists of three components: red, green, and blue. The RGB components vary from 0 (black) to 255 (white).

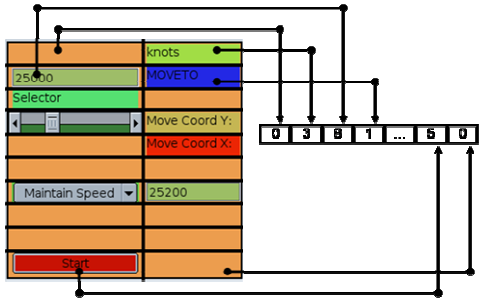

Figure 5. Encoding of the widget layout: Widgets are identified by integer IDs (>0) and empty cells in the grid are identified with 0s

Genetic Operators

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 6.

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*).

The widget layout chromosome is an integer sequential encoding, and in order to preserve this permutation representation, we use the Partial Mapped Crossover (PMX) operator. PMX keeps crossover from creating individuals with duplicate genes, which would violate the permutation property. We also use swap mutation, which randomly picks two alleles from the chromosome and swaps them. For the widget style chromosome, we use single point crossover and bit-flip mutation operators.

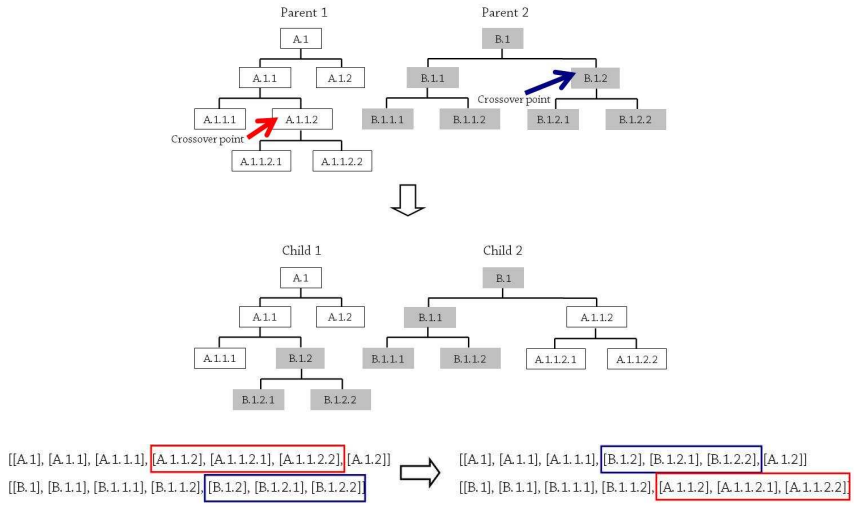

Figure 6. Tree crossover operator

Fitness

There are times when it is difficult if not impossible to determine the fitness function for a problem domain when using evolutionary computation. An IGA replaces the fitness evaluation with the user. IGAs are useful when there is no better fitness measure than the one in the human mind. IGAs have been applied to various domains, ranging from artistic and highly creative applications to engineering [10].

A typical IGA session consists of a user evaluating a set of individuals from the IGA population. The individuals of the population are then assigned a fitness value (or values), based on subjective and objective crite-

ria of evaluation. In the floorplanning problem, we separate objective criteria from subjective criteria as, (1) the only measurable objective in floorplans is their compliance with the Architect data guidelines [22]. The guidelines for single-storey house plans relate to minimum room dimensions and areas. Every individual floorplan is assigned a fitness value based on its compliance with the minimum dimension and minimum area guidelines. (2) The IGA lets the designer pick a particular floorplan as being the "best". This subjective pick (based on preferences) is translated into his or her preference for the number of rooms, total built area (area occupied by *rooms*), and room adjacencies. An individual plan is compared with the "best" plan and assigned high fitness values if the plan is similar to the "best" plan in each of the three subjective criteria.

The objective component of fitness for evolving widget layout and style comes from UI style guidelines. The main guideline currently incorporated in the objective evaluation is the use of highly contrasting background and foreground colors. The grid positioning of the widgets in the layout automatically enforces a widget alignment guideline. The objective fitness is the Euclidean distance between the color vectors of the foreground and background colors. The subjective evaluation consists of finding the similarity between the currently evaluated individual with the user selected best by using the longest common subsequence (LCS). We find the length of the LCS of the layout chromosome (length1) and of the style chromosome (length2). We add these two lengths and use the sum as the subjective fitness score.

Genetic Algorithm

We use the Non-dominated Sorted multi-objective Genetic Algorithm, abbreviated as NSGA-II [23]. The NSGA-II creates fronts of non-dominated individuals, where within a front none of the individuals are any worse than any other individual across all optimization criteria. All individuals within a front are said to have the same rank. We select parents by using the crowded distance tournament operator. We pick two individuals to participate in the tournament, and we select the individual with the higher rank to be part of the mating pool. In case the two individuals have the same rank, and consequently belong to the same front, then the crowded distance of both individuals is computed, and we select the individual with the highest crowded distance to be part of the mating pool. This translates to the individual being in a less crowded region of the front and hence, the crowded distance selection favors the most diverse individuals within a front.

We implement both the floorplans design and widget layout/style evolution with the NSGA-II. For floorplans design we use a four-criterion multi-objective minimization function. With the widget layout/style implementation we keep the objective and subjective criteria separate and use a two-criterion minimization function for NSGA-II. We have also used the standard canonical GA where we combine the subjective and objective fitness into a single weighted linear sum [24].

Visualization of Solution Space

We display a subset of nine individuals, from a large population size, to be evaluated by the designer. We chose nine because it allows us to display a visually appealing grid of 3x3 individuals, which also does not overwhelm the designer. What to display from the population to be evaluated by the user is a critical step, since displaying useful information to the user makes for a productive session, while displaying a poor subset can inhibit the progress of the interactive evolutionary process.

Various methods of selecting a small subset from a large population have been previously explored [24-26]. In the work presented in this paper we select individuals from the fronts created by the NSGA-II. We pick three individuals from the first front, three individuals from the second front, and so on until we have obtained nine individuals. If there are less than three fronts, or if one of the fronts has less than three individuals, then we obtain the next three individuals from the next front in a round robin fashion. 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 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. A snapshot of the interactive screen for non-collaborative floorplanning is shown below in Figure 7.

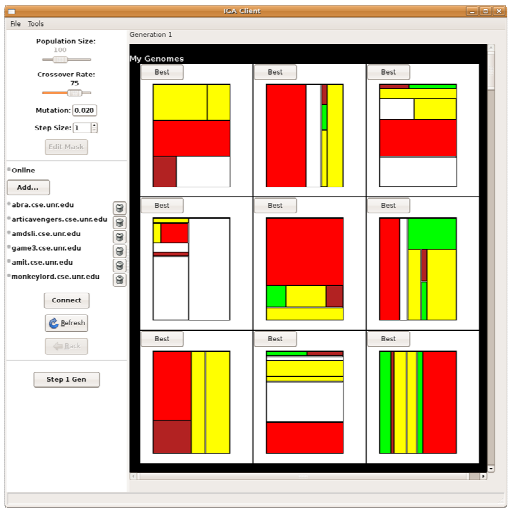

Figure 7. The non-collaborative interactive interface for floorplanning

The user input consists of selecting the individual the user likes the best from either the subset, or from one of the individuals from the rest of the population (viewed by scrolling). We use the user selected best to interpolate the fitness of every other individual in the population. On the top of each individual displayed, we add a button with the label "Best". By clicking on the "Best" button of a design displayed to the user, he or she provides input to the IGA regarding the fitness criteria. Currently, we only support for only one individual to be selected as the best. Through the interface we also support the ability to provide input every n^{th} generation, where the value of n stands for 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 rates through the interface.

Collaborative Framework

The collaborative module is wrapped over the interactive module and is what binds the individual IGA sessions together. Collaborative evolution is implemented by networking with a peer to peer network. We treat each user participating in evolution as a node, handling incoming requests from

other nodes (peers) and requesting information from peers. By using a peer to peer network, control is decentralized and each node is free to chose who to connect to and if necessary who to exclude from its set of peers. Note that connections between peers must be direct, we do not support for spidering connections, where node-A can connect to node-C through node-B. Since each node consists of a server to handle request from any peer, each node can broadcast its signal to any peer that connects to it.

Collaborative Interface

During collaborative evolution, a subset of peer-evolved designs is displayed to the right of the user's population. We limit the number of peer individuals to nine, organized in a 3x3 grid, similar to how we present the user's own population, in order to be consistent. For more than one peer, we cannot display all the individuals belonging to the subset of each peer, since we only display nine. We do make sure that the user selected best individuals from each peer are displayed on the peers subset. We save the user selected best from generation to generation, and we always make it part of the subset displayed the next time the IGA requires user input. The reason for making sure that a peer can see the user selected best from other peers is that if a user selects an individual as the best, then it was because the user found the selected individual to be the most interesting/intriguing/creative and to be the best candidate to bias the evolution of his or her own population, as well as those of other users. We select the rest of the individuals that make up the peers subset by taking a random subset from a collective pool of all individuals that make up peers' subsets. By selecting a random subset, we believe that over many generations, all of the participants will get approximately the same amount of their designs displayed on the screens of collaborators.

The benefit of viewing the best individuals from peers is limited, unless the user is able to take promising individuals from peers and mold them to their liking. We support this by allowing the user to inject individuals from the subset of peers into the user's own population. The user can select an individual from a peer to be added to the user's own gene pool by clicking on the "Add to Genome" button. The user can also select a best individual from the subset of individuals from peers, in which case the user selected best is automatically injected into the population, and used for fitness interpolation. We require the user to select a best individual, but it does not have to be from the user's own population – the user selected best can come from peers. The collaborative interface is shown in Figure 8.

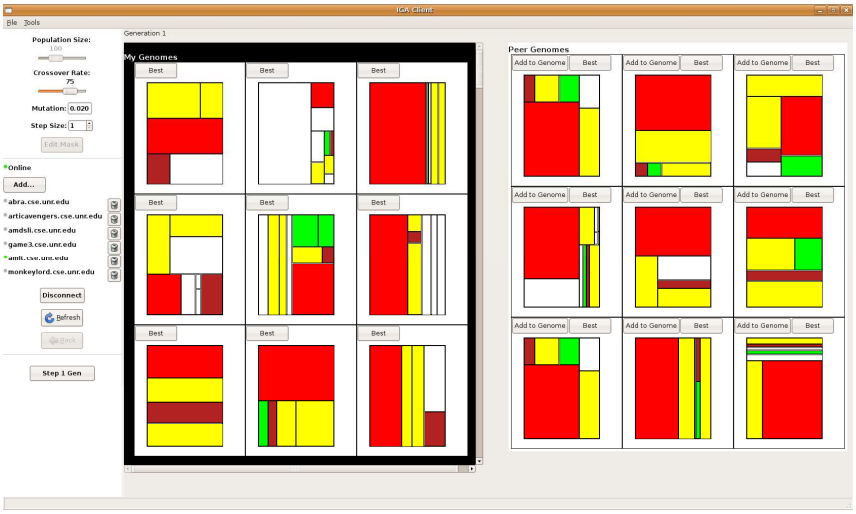

Figure 8. The collaborative interactive interface for the floorplanning problem

Fitness Bias

We use fitness biasing to ensure that injected individuals survive long enough to leave a mark on the host population by using the concept of bloodline. Injected individuals are considered to be *full blood*, while those individuals already in the population are treated as individuals 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 individuals 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 [27]. 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.

Computational Results

We present some preliminary results for the floorplanning and widget layout/style design problem. Floorplans were evolved based on a simulated design brief that stated the problem as a minimally constrained one. The

brief was 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 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 by the authors and two of their colleagues. During the collaborative evolution, only nine representative designs were made visible to every designer in the peer-group. Every designer also had access to a subset of nine evolving designs from the populations of the four other peers. However, during individual evolution of floorplans for the same problem, the designer had visual access to all the evolving designs in his own population to ensure some sort of parity in the visualization space vis-à-vis the collaborative evolution. In other words, for a standard population size of 100, the designer participating in the collaborative evolution effort had visual access to 18 designs at a time, while the same designer involved in the individual pursuit of evolution had visual access to all 100 designs.

We also used a design template to aid us with the first fitness computation. For a population size of 100, the initial population has 99 floorplans that are created randomly and the design template constituted the 100^{th} floorplan. The template serves as the assumed "best" for the first generation, and the randomly created floorplans are compared to the template and assigned the three subjective fitness values discussed in section 4.1.3. All floorplans are also evaluated based on their compliance with the Architectural data guidelines. Based on the four criteria, the Pareto-optimal fronts are calculated and three members selected at random from the first fronts each are displayed to the designer. The designer then interacts with the interface and brings the evolution to a stop when he or she feels that there are one (or more) *interesting* designs in the current population.

Solutions to the two-bedroom one-bathroom problem described earlier were evolved over multiple runs both individually and collaboratively. Interestingly, we used as a design template a one-bedroom one-bathroom plan, which goes on to show that the choice of template is not an issue (the template itself was lost in 5-6 generations). We achieved satisfactory results (floorplans meeting all or most of the constraints) in most cases within 15-20 generations. In Figure 9 a set of six floorplans the designers considered interesting while individually evolving floorplans is shown, and in Figure 10 we show a set of six floorplans that the same designers considered interesting when they collaborated on the same problem. The rooms are color coded as red (living area), yellow (bedrooms), green (eating areas – kitchen and/or dining rooms), firebrick (bathrooms) and white (empty spaces in the plan).

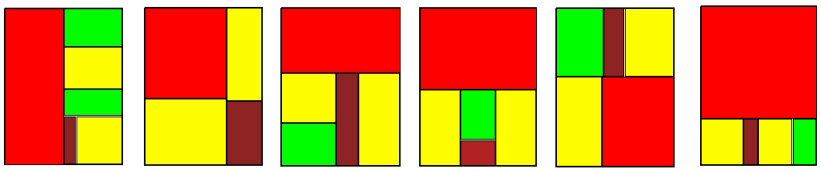

Figure 9. Six representative floorplans for the design problem evolved non-collaboratively

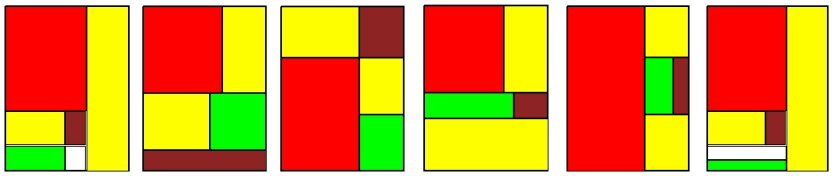

Figure 10. Six representative floorplans for the design problem evolved collaboratively

The plans were evaluated for creative content based on practicality and originality on a five-point scale [28]. We conducted this as an initial pre-test that would eventually help us devise a more effective methodology for design concept evaluation. We sought the participation of ten graduate students, five from our lab and five outside of our lab. In addition to the two viewpoints of practicality and originality, we also asked survey participants to rate how close a design comes in meeting the minimum design requirements and the set of constraints. Based on this, it was unanimously felt that #5 (Figure 9) and # 9 (Figure 10) did not meet the north-west living room constraint and hence, it was decided to omit them from the evaluation. The results of the evaluation are presented in Table 1.

The collaboratively designed floorplans were consistently rated higher on the originality scale, while the individually (non-collaborative) designed floorplans were found to be more practical. Many survey participants ranked #12 (Figure 10) as being the most original and the least practical floorplan. Because of the limited size of the survey population, it would be premature to conclude that collaboratively developed floorplans do not address resolution aspects as well as the individually developed floorplans do. We also feel that a designer-centric viewpoint of creativity should hold as much weight as peer-evaluation based on practicality and originality. Design #7 (Figure 10) came about when one designer evolving

designs with “two-bedrooms one-bathroom with no eating areas” (similar to #2) injected his population with a peer’s design, one who was evolving plans with a kitchen area (similar to #3). Neither designer was likely to come up with #7 on their own, which also happens to score highly on the originality scale. In fact, all the collaboratively designed floorplans were rated in the top-5 on the originality scale.

Table 1 Creativity evaluation of the ten representative design concepts

DESIGN #	PRACTICALITY	ORIGINALITY	RANK
1	2.7	3.1	6
2	2.9	2.7	8
3	2.4	3.3	7
4	**4.4**	3.3	2
6	4.0	3.2	3
7	2.2	3.4	8
8	2.8	3.4	5
10	3.2	3.5	4
11	4.2	3.6	**1**
12	1.6	**3.8**	10

The UI panel evolution shows promise, but further experiments and data analysis are required. Figure 11 (left) shows three UI panels evolved individually and Figure 11 (right) shows three panels evolved collaboratively. Although the preliminary results are inconclusive, we did find that collaboratively evolved panels, showed overall visually appealing softer color tones, while individually evolved panels show a high degree of contrast between background and foreground color (at times uncomfortable combinations).

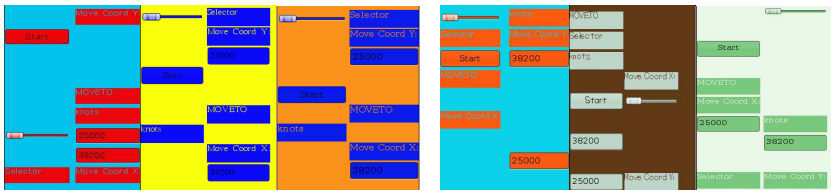

Figure 11. Left: Three representative UI panels evolved individually (notice the high contrast between foreground widget colors and background panel color); Right: Three representative UI panels evolved collaboratively

Conclusions

In this paper, we propose a collaborative model for creative designing based on interactive genetic algorithms. We implement our proposed model to collaboratively evolve floorplans and widget layout designs – two applications that may have potential use in interactive development of user-interfaces. We have addressed several issues relating to implementation and issues relating to genetic search, interaction and collaboration. We also compare results of collaborative evolution with similar results obtained with individual (non-collaborative) evolution, by evaluating created designs on a five-point scale for practicality and originality. The pre-test indicates that while individually created floorplans were rated highly for practicality, the collaboratively generated floorplans were considered more original. Based on these preliminary findings, we believe that there is enough empirical evidence to support our hypothesis that collaborative interactive evolutionary search of design spaces is indeed a viable computational model of creative design.

Acknowledgements

We thank the survey participants for their time. 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

1. Renner G, Ekrárt A (2003) Genetic algorithms in computer aided design. Computer-Aided Design 35: 709-726
2. Bentley PJ, Wakefield JP (1997) Conceptual evolutionary design by genetic algorithms. Engineering Design and Automation Journal 3: 119-131
3. Goldberg DE, Rzevski G (1991) Genetic algorithms as a computational theory of conceptual design. Applications of Artificial Intelligence in Engineering VI: 3-16
4. Bentley PJ, Corne, DW (2002) An introduction to creative evolutionary systems. In Creative Evolutionary Systems. New York: Academic Press
5. Rosenman MA (1997) An exploration into evolutionary models for non-routine design. Artificial Intelligence in Engineering 11: 287-293
6. Rosenman MA (1997) The generation of form using an evolutionary approach. In Evolutionary Algorithms in Engineering Applications. Berlin Heidelberg: Springer-Verlag

7. Gero JS, Schnier T (1995) Evolving representation of design cases and their use in creative design. In Proc 3rd Int Conf Comput Models Creative Design
8. Poon J, Maher ML (1997) Co-evolution and emergence in design. Artificial Intelligence in Engineering 11: 319-327
9. De Silva Garza AG, Maher ML (2000) Characterising evolutionary design case adaption. In Artificial Intelligence in Design. Dordrecht: Kluwer Acad
10. Takagi H (2001) Interactive evolutionary computation: fusion of the capabilities of EC optimization and human evaluation. Proc IEEE 89: 1275-1296
11. Wilson P (1991) Computer supported cooperative work: An introduction. Dordrecht: Kluwer Acad
12. Peng C (2001) Design through digital interaction: Computing communications and collaboration on design. Intellect Books
13. Csikszentmihalyi M (1997) Creativity: Flow and the psychology of discovery and invention. Harper Perennial
14. Boden M (1999) Computer models of creativity. In Handbook of Creativity, Cambridge: Cambridge Univ Press
15. Suwa M, Gero J, Purcell T (2000) Unexpected discoveries and S-invention of design requirements: Important vehicles for a design process. Design Studies 21: 539-567
16. Csikszentmihalyi M (1988) Society, culture and person: A systems view of creativity. In The Nature of Creativity. Cambridge: Cambridge Univ Press
17. Campbell DT (1960) Blind variation and selective retention in creative thought as in other thought processes. Psych Rev 67: 380-400
18. Simonton DK (1999) Creativity as blind variation and selective retention: Is the creative process Darwinian. Psych Inquiry 10: 309-328
19. Martindale C (1995) Creativity and connectionism. In The Creative Cognition Approach. Cambridge, MA: MIT Press
20. Woodbury RF, Burrow AL (2006) Whither design space. Artificial Intelligence for Engineering Design, Analysis and Manufacturing 20: 63-82
21. Gero JS (2002) Computational models of creative designing based on situated cognition. In Proc 4th Conf Creativity Cognition: 3-10
22. Neufert E, Neufert P, Baiche B, Walliman N (2002), Architects' data. Wiley
23. Deb K (2001) Multi-objective optimization using evolutionary algorithms. New York: John Wiley
24. Quiroz JC, Dascalu SM, Louis, SJ (2007) Human guided evolution of XUL user interfaces. In Computer Human Interaction. San Jose, CA: ACM Press
25. Lee JY, Cho SB (1999) Sparse fitness evaluation for reducing user burden in interactive genetic algorithm. In Proc Int Fuzzy Syst Conf 2: 998-1003
26. Quiroz JC, Dascalu SM, Louis, SJ (2007) Interactive evolution of XUL user interfaces. In Proc Conf Genetic and Evolutionary Comput: 2151-2158
27. Louis SJ, Miles C (2005) Playing to learn: case-injected genetic algorithms for learning to play computer games. IEEE Trans Evol Comput. 9: 669-681.
28. Finke R, Ward T, Smith S (1992) Creative cognition: Theory, research and applications. Cambridge, MA: MIT Press