Supporting Free-form Design Using A Component Based Representation: An Overview

Azahar Machwe
ACDDM Group, UWE, Bristol
Frenchay Campus, Coldharbour Lane
Bristol, BS16 1QY, UK
+44 (0)117 328 3137
azahar.machwe@uwe.ac.uk

Ian C Parmee
ACDDM Group, UWE, Bristol
Frenchay Campus, Coldharbour Lane
Bristol, BS16 1QY, UK
+44 (0)117 328 3137
ian.parmee@uwe.ac.uk

ABSTRACT
This paper reviews the development of an interactive evolutionary design system (IEDS) for conceptual design, which integrates an agent based 'build and evolve' approach with a component based representation. It builds upon the previous work done by the authors involving interactive design of bench-like structures and describes the extension of the system to handle the design of free-form urban furniture. The extensions include integration of solution clustering and a free-form design rule-base, within the IEDS. The presented results highlight the various advantages of combining an agent based solution assembly method with a component based representation.

Categories and Subject Descriptors
J.m [**Miscellaneous**]

I.2.m [**Artificial Intelligence**]: Miscellaneous

General Terms
Design, Algorithms.

Keywords
Interactive Evolutionary Design, Software Agents, Aesthetics

1. INTRODUCTION
The conceptual design stage of any design process is characterized by limited knowledge about the design problem and high flexibility in the choice of solutions. As the design progresses from conceptual design stage to detailed design it gets transformed from a collection of abstract concepts to a well understood system [1]. During the conceptual design stage the cost of incorrect design decisions and the problems in correcting them can be managed easily. The costs of correcting design flaws increases rapidly as we move towards detailed design stage. The emphasis during conceptual design stage is not on producing a detailed design solution but on understanding and subsequently reformulating the design problem [2], [3] and to obtain a set of possible candidate solutions which can then be analyzed at a higher level of detail during the later design stages. Therefore during the conceptual design stage importance should be given to search and exploration of the design space rather than the search for optimum solutions.

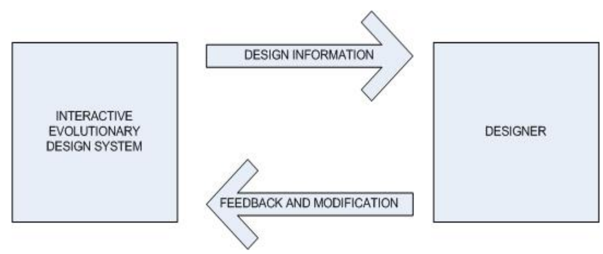

Figure 1: The interactive evolutionary design loop.

Keeping the above factors in mind, any interactive evolutionary design system (IEDS) for supporting conceptual design should provide a flexible platform for rapid solution generation and analysis. Flexibility is required in terms of the variety of solutions which can be represented within the system. This requires the solution representation used within the IEDS to be flexible enough to represent a wide range of solution types.

The integration of the designer within the interactive design loop (see Figure 1) is also an important factor to keep in mind while implementing an IEDS for conceptual design. This is so because during the conceptual design stage, design activity is still based around human designers and requires them to draw on their experience and skills to come up with novel solutions to design [4] problems. Furthermore no machine based design function can replace the judgment of an experienced and skilled designer especially when evaluating the solutions subjectively (for example visual aesthetics in case of an automobile).

2. BACKGROUND
When it comes to interactive evolutionary design systems, we find a wide variety of application domains. For a detailed review of various application domains the reader is directed to [5].

While developing an IEDS for conceptual design, the main focus of the authors, was on combining a flexible design representation, software design agents with a mixed (user-based and machine-based) fitness assessment of the solutions to provide machine based support to the designer.

GECCO'07, July 7-11, 2007, London, England, United Kingdom.

Figures 2a-c show the various versions of the IEDS and how it grew from a collection of simple components (Figure 2a) into its current state (Figure 2c). The initial system (Figure 2a) consisted of three main interacting components: the user, the software agents [6] and

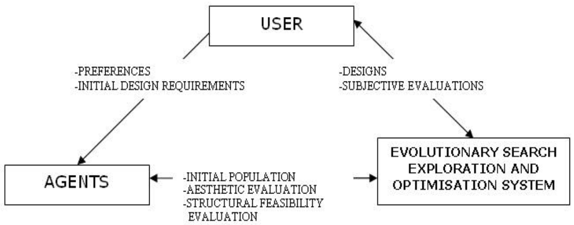

Figure 2a. Initial, high level design of the IEDS.

the evolutionary search and exploration system. The software agents were responsible for providing construction and repair services (required for the build and evolve approach) as well as the fitness assessment of the solutions.

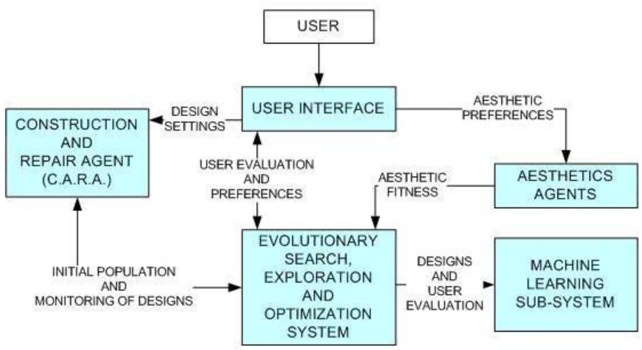

Figure 2b. Expanded high level design of the Interactive Bridge Design System.

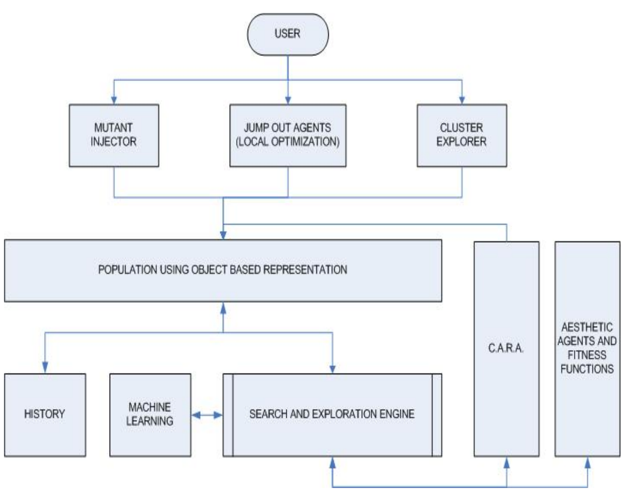

Figure 2c. Expanded high level design of the Interactive Urban Furniture Design System.

The evolutionary system used was a simple evolutionary programming (EP) algorithm [7] since we wanted a relatively simple search algorithm to have maximum flexibility in design of the IEDS.

The next design increment, as used for the Interactive Bridge Design System (see Figure 2b) [6], added a machine learning sub-system to the IEDS and divided the software agents into

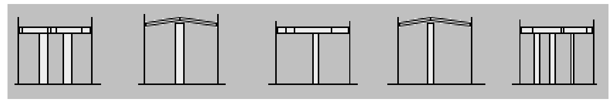

Figure 3. Some optimized bridge designs.

Construction and Repair Agent (C.A.R.A.) and Aesthetics Evaluation Agent. The designs generated (see Figure 3) by the bridge design system were 2-D side views of simply supported bridges. The nature of the design problem and the analysis of solutions were kept simple since the system was always envisaged as a proof of concept prototype. For further details of the bridge design system the reader is directed to [6].

The final design increment, as used for the Interactive Urban Furniture Design System [8] added various other components to the IEDS such as: Clustering, Jump Out Agents for local optimization, a Mutant Injector and a Solution Store (History). The C.A.R.A. was

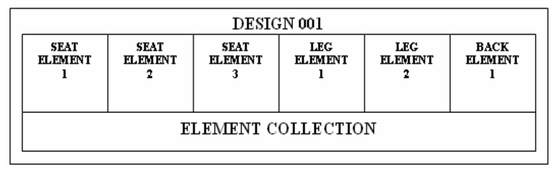

Figure 4. An example of a Bench solution.

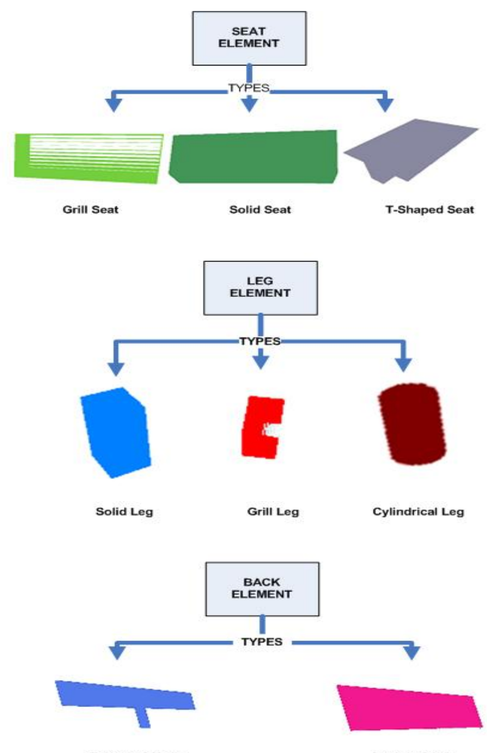

Figure 5. An example of different Element styles (or types).

expanded further and provided with a flexible rule base to generate free form as well as rigidly defined initial

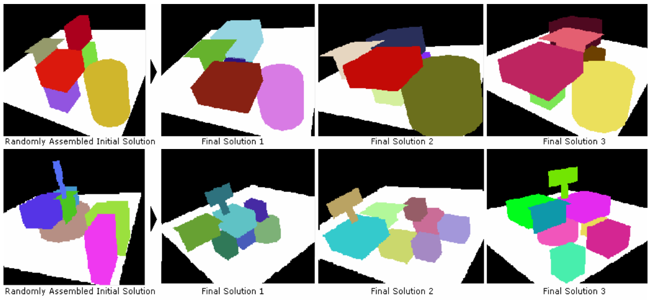

Figure 6a. Initial solutions (extreme left) and three evolved results using a flexible rule base.

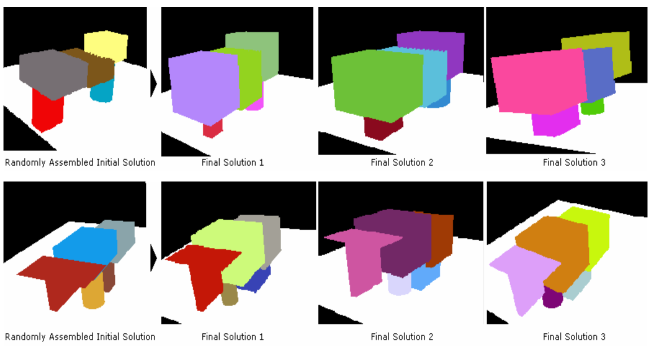

Figure 6b. Initial solutions (extreme left) and three evolved solutions using a rigid rule based.

solutions. It also increased the design complexity and added a rich set of components with different styles and structural behavior. The Urban Furniture Design System initially concentrated on the design of well defined ‘bench-like’ structures but in a later version, it was extended to the design of free-form seating arrangements. This paper reviews the free-form version of the urban furniture design system.

3. REPRESENTATION AND AGENCY

The component based representation used within the Bridge Design system was extended to 3-D space within the initial version of the Interactive Urban Furniture Design System [8]. The representation was initially used with a very well defined set of construction rules. The designs were based loosely around a standard bench configuration. This allowed extensive engineering analysis of the solutions including the calculation of cantilever and simple beam deflection values for the structure. For further details about the representation and the initial version of the Urban Furniture Design System, the reader is directed to [9].

After extensive user based testing of the system the flexibility provided by the component based representation and rule based assembly method was used to create and evolve free-form designs. Simple construction rules were used and the number of component types was also increased. An example solution can be seen in Figure 4. It has three seat element components, two leg element components and one back element component. The list of component types can be seen in Figure 5.

The Construction and Repair Agent consists of two distinct components, specifically the Construction Agent (CA) and the Repair Agent (RA). The Construction Agent is responsible for the assembly of the initial population using a set of rules and for ensuring that none of the assembled solutions violate the design

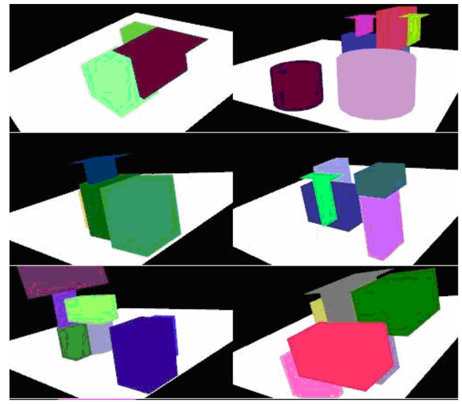

Figure 7a. Example of an initial set of cluster representatives.

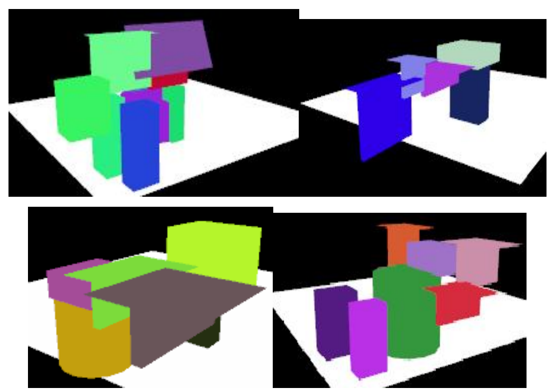

Figure 7b. Some resulting solutions (after 50 generations).

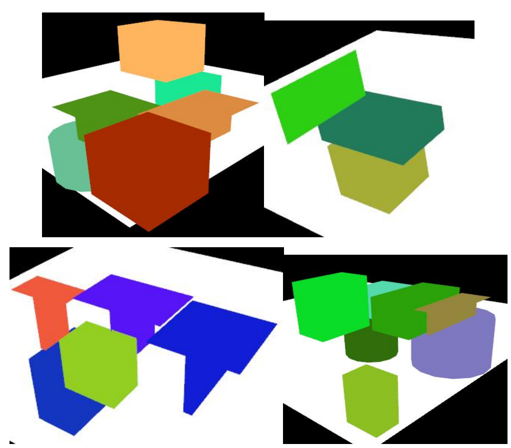

Figure 8: Some user evolved solutions.

constraints imposed by the designer. The Repair Agent is responsible for ensuring that the solutions continue to remain within constraints throughout the evolutionary process since exploratory operations (such as mutation) can disrupt the designs. Since both the Agents use a set of rules to create and maintain solutions, the more flexible the rules the greater will be the variety of designs generated by the system.

Figures 6a and 6b show the impact of the nature of the rule-base on the initial and final solutions for a few typical runs. The free-form rule-base (Figure 6a) results in solutions with different configurations. Upon comparing it with the 'rigid' rule-base (Figure 6b) it can be seen that, while there are structural changes between the assembled and evolved solutions, the overall configuration of the design remains the same.

4. CLUSTERING

Within the Bench design system we utilize clustering based on the number of different component types present within a solution. For example all solutions having a combination of three seat, two leg and one back element would be the part of the same cluster. Clustering is not used for the purpose of fitness assignment [10] within the system.

Clustering has the dual advantage of allowing the use of a much larger population size than usually possible within interactive evolutionary systems and improving the presentation of solutions to the user. For details of the process and results the reader is directed to [9].

5. RESULTS

The system uses a simple evolutionary programming algorithm with a rule based mutation operator. Tournament selection is used as the selection mechanism within the algorithm. Fitness is a combination of rule based aesthetics, user assigned fitness, structural continuity analysis and seating area analysis.

5.1 With Machine Based Fitness Function

The system was run for fifty generations with a tournament size of three. The population size was set at N=40. The solutions were evaluated using only the machine- based fitness function. Figure 7a shows an example of the cluster representatives of the initial population.

Figure 7b shows some machine evolved solutions, over multiple runs, after fifty generations. It can be seen that a wide variety of shapes and configurations result when a relatively free form set of rules is utilized. It is, in fact, quite difficult to locate a pattern in the evolved solutions but a pattern does exist. The increased seating area and continuity of the final solutions reflect the fitness measure of structural continuity and seating area. Another interesting feature to note is the appearance of spread out seating structures containing a leg element and a seat element each. Such structures arise when a large continuous bench with multiple leg elements, breaks down into smaller structures.

5.2 With User Evaluation and Machine-Based Fitness Function

The population size and tournament size remains the same as before. The only difference is that, the user is also evaluating the solutions along with the machine-based fitness function. Experiments were run with four users working independently on the urban furniture design system. Some of the user-evolved high fitness solutions are shown in Figure 8. In case of user-interactive runs, the final solutions have a wide variety of configurations including a traditional 'bench-type' configuration which is absent in the fully machine-based runs.

6. CONCLUSION

In this review paper a brief description of the urban furniture design system has been provided. The novel features such as the clustering and representation were highlighted.

The presented results verify the stated advantages of such a combination of agency based build and evolve services along with a component based representation. It was shown, that by just changing the assembly rule base and the fitness function a wide variety of solutions can be obtained. During the conceptual design stage such a flexible representation can be used with different rule-sets to provide a whole spectrum of designs for the designers to work with.

7. REFERENCES

[1] L. Xu, Z. LIy, S. LIy and F. Tang (2005) A polychromatic sets approach to the conceptual design of machine tools International Journal of Production Research, Vol. 43, No. 12, 15 June 2005, 2397–2421

[2] Parmee, I.C. (2002). "Improving problem definition through interactive evolutionary computation". Artificial Intelligence for Engineering Design, Analysis and Manufacturing (2002), 16(3), pg. 185-202. Cambridge University Press, Printed in USA.

[3] Goel A. K., (1997), Design, Analogy and Creativity. IEEE Expert, Intelligent Systems and their Applications, 12(3). pp 62 -- 70.

[4] Wood W.H. and Agogino A.M. (1996) A Case-Based Conceptual Design Information Server for Concurrent Engineering, Computer-Aided Design, Volume 28, Number 5, May 1996, pp. 361-369(9)

[5] Takagi, H. (2001). Interactive Evolutionary Computation: Fusion of the Capacities of EC Optimization and Human Evaluation. *Proceesings of the IEEE 89, 9,* pp. 1275-1296

[6] Machwe, A. Parmee, I.C. and Miles, J.C. (2005) Integrating Aesthetic Criteria with a user-centric evolutionary system via a component based design representation. Proceedings of International Conference on Engineering Design (2005), Melbourne, Australia.

[7] Fogel, D.B. (1988). An Evolutionary approach to the travelling salesman problem. Biological Cybernetics 60(2), 139-144.

[8] Machwe, A. and Parmee, I.C. (2006) Integrating aesthetic criteria with evolutionary processes in complex, free-form design – an initial investigation. Proceedings of the IEEE Congress on Evolutionary Computation, 2006, Vancouver, Canada.

[9] Machwe, A. and Parmee, I.C. (2007) Towards an interactive, generative, design system: Integrating a 'build and evolve' approach with machine learning for complex freeform design. Evo-INTERACT 2007, Valencia, Spain

[10] H.S. Kim and S.B. Cho. An efficient genetic algorithm with less fitness evaluation by clustering. pages 887{894. Proceedings of the 2001 IEEE Congress on Evolutionary Computation, Seoul, Korea, 2001