John S. Gero, Scott Chase and Mike Rosenman (eds), *CAADRIA2001,*
Key Centre of Design Computing and Cognition, University of Sydney, 2001, pp. 137-148.

GENERATING-PREDICTING SOUP

A conceptual framework for a design environment

PATRICK H. T. JANSSEN, JOHN H. FRAZER AND MING X. TANG
The Hong Kong Polytechnic University
Hong Kong, China

Abstract. A framework for the development of a computational environment that supports generative design is proposed. This environment is referred to as a generating-predicting soup. This paper discusses such an environment at a conceptual level. The research focuses on the architecture, engineering, construction, and facilities management (AEC/FM) domain. The general framework is however applicable to many design domains.

1. Introduction

Designing involves imagining and foreseeing future possibilities. Materialisation and deployment put these possibilities into action. The imagining aspect of designing relies upon a certain level of personal idiosyncratic input in order to feed the creative process. The foreseeing aspect relies on the ability to make rational predictions about how the possibility might be materialised and deployed. A computational environment that aims to enhance these two aspects of designing is proposed. This environment is referred to as a generating-predicting soup (G-P Soup).

The imagining aspect is enhanced by a generating process. Such automatic generation of design proposals by computers is a highly active area of research. Many different approaches have been proposed, resulting in diverse systems. However, many of these systems tend to be overly focused on finding the generic 'optimal solution' rather than allowing for the development of 'my proposal'. The generating process aims to welcome personal idiosyncrasy by allowing a particular style and methodology to be defined.

The foreseeing aspect is enhanced by the predicting process. Often, what is foreseen does not match up with what actually occurs during materialisation and deployment. This disparity can manifest itself in a wide variety of forms. Sometimes these consequences are expensive. At other times they may contribute to social or moral decline. Still other times the consequences may involve catastrophic collapse. The predicting process aims to minimise this disparity by using sophisticated analysis and simulation software applications in order to predict possible outcomes.

2. The Generating - Predicting Soup

The G-P Soup is visualised as 'soup' within which design proposals 'float' and 'sink'. New design proposals are continuously being dropped into this soup and old ones are continuously sinking. A dynamically stable population of floating design proposals is maintained. Three processes are fundamental: the generating process and the predicting process, and the sinking process. The generating process continuously drops new design proposals into the soup. The predicting process continuously makes predictions about how suitable these new design proposals might be. The sinking process maintains a stable population by ensuring that the least suitable design proposals sink. Each of these processes occurs asynchronously. Together, these processes ensure that the floating design proposals will gradually evolve over many generations, resulting in highly complex and adapted design proposals.

2.1 THE GENERATING PROCESS

The generating process requires the encoding of domain knowledge at two levels of interpretation: the generic and the specific. The generic interpretation of a particular domain is encoded as an overall representational framework. The specific interpretation reflects the ideas of a particular designer or design office. This interpretation is encoded as a highly adaptable design prototype together with a set of associated generative rules. This adaptive prototype is referred to as a 'formative'.

The representational framework is an encoding of the common generic concepts in a particular domain. Relevant concepts for many design domains include the processes and products involved in the imagining, foreseeing, materialisation and deployment of an idea. The representational framework will encode these concepts as a set of entities and relations. Due to its generic nature, the representational framework is applicable to many projects and even to many designers or design studios.

The formative and the associated generative rules encode concepts relating to a design stance and methodology. The definition of the formative will rely heavily on the entities and relations defined by the representational framework. However, idiosyncratic concepts that do not exist in this generic framework can be defined within the prototype. From this formative, a wide variety of design proposals can be generated that differ from each other in overall structure and organisation as well as in detail. Such design proposals are generated by applying the generative rules to the formative. The rules are applied in a certain sequence, one after another, gradually transforming the formative.

In order to generate a new design proposal, the generating process must decide which rules to apply and in what sequence. This information is extracted from the design proposals that are found floating in the soup at that moment in time. Each of these design proposal carries with it a historical list of the rules that were used to create it. The generative process randomly selects a number of design proposals, extracts their rule lists and uses 'cut and paste' type mechanisms in order to construct a new list. This list is then used to generate a new design proposal that is then dropped into the soup.

2.2 EVOLUTION

The gradual evolution of the design proposals being produced, from primitive and naïve to highly adapted and complex, is a fundamental and emergent quality of the G-P Soup. Without this, the proposals produced would be of little value. Such an evolutionary process can be attributed to three key factors: selection, transmission and variation (Dawkins 1983). In the G-P Soup, selection exists as a result of the sinking process. The least suitable design proposals sink, ensuring that the most suitable are selected to stay afloat. Transmission exists as a result of the method by which new design proposals are created. The generating process creates these new design proposals by 'cutting and pasting' the generative rules of a number of 'selected' design proposals. Variation exists as a result of small random changes introduced during the transmission. Perhaps two rules are swapped, or a rule is deleted.

2.3 THE ROLE OF THE USER

The G-P Soup is envisaged to be a highly interactive environment; an impressionable ecosystem that is susceptible to manipulation and intervention. Thus, users of the system are able to give preference to certain design proposals over others, save some design types from extinction, change prediction criteria in mid run, tweak formatives, and so forth. The aim is not to achieve converge on a small set of optimal solutions. Rather, the aim is to allow the user to muse upon the diverse design proposals to be found in the soup at any one time.

The G-P Soup intentionally demands its users to have new skills. Robert Aish of Bentley Systems calls this new skill "end user programming" for "computer based design tools as (visual) programming environments". Aish (2000) writes that by “providing tools to allow non-programmers to create new applications, we may (deliberately) lower the skill requirements for 'end-user programmers' and effectively blur the distinction between the 'end-user programmer' and the final 'end user', and a decrease in the complexity of the skills required to be a 'programmer' end-user”. Similar types of systems, relying on 'designer-programmer' end users, are seen to be applicable to the G-P Soup. Thus, formatives may be designed in visual 'drag-and-drop' environments. Generative rules may be designed utilising intuitive 'write-and-test' interfaces that allow fine tuning.

2.4 KEY CONCEPTS

A fundamental idea behind the G-P Soup is that designing is a synthesis of personal idiosyncrasy and rational foresight. These two concepts from that basis of this environment. The generating process relies on personal idiosyncrasy. The predicting process results in rational foresight.

Section three will discuss the concept of personal idiosyncrasy in design. Section four will discuss the concept of rational foresight in design.

3. Personal Idiosyncrasy

3.1 THE ONE-OFF MIND

In an article in The Guardian newspaper Nicholas Humphrey (1987) compares the value of ‘discoveries’ in the scientific realm to ‘creations’ in the artistic realm. Humphrey argues that if you were forced to 'consign to oblivion' either Newton's *Principia Mathematica* or the Eiffel Tower, then the *Principia* would have to go. Why? If Newton had not written it then someone else would have done. "The *Principia* was a glorious monument to human intellect, the Eiffel Tower was a relatively minor feat of romantic engineering; yet the fact is that while Eiffel did it his way, Newton merely did it God's way".

One may question whether ‘scientific discoveries’ are really so inevitable. However, what is not questionable is that ‘artistic creations’ are definitely not inevitable. If Eiffel had not existed, then we can be sure that the Eiffel Tower would not have existed either. This individualistic quality is true, to a certain extent, of all design. This is not to say that design is free from objective criteria. The Eiffel Tower had many objective criteria; it had to be structurally stable, it had to be very tall, it had to be constructable with the technology and materials then available, and so forth. Nevertheless, the idea of an Eiffel Tower was specific to Eiffel himself.

3.2 THE SEARCH METAPHOR IN DESIGN

Personal idiosyncrasy brings into question another characterisation of design that is particularly prevalent within the field of design automation. This characterisation describes design as a search problem and may be traced back to Newell, Shaw and Simon (1957) paper entitled *Elements of a Theory of Problem Solving*. The information processing model that was proposed in this paper sought to explain problem solving behaviour through basic information processes. Such information processing models led to a whole field of research into the cognitive basis of designing. From this information processing perspective, design was seen as process of searching for solutions to design problems. That search might play a role within the overall process of designing is not disputable. Algorithmic problem solvers and solution optimisers are valuable tools for modern day design. What is disputable is whether a search based paradigm is helpful in describing the overall process of designing.

The idea that there might exist a single optimal solution to a certain design 'problem' has already been countered by what might be described as Humphrey's oblivion argument. Many researchers in the field of design have highlighted various incompatibilities between design and search (Janssen 2001). For example Horst Rittel (Churchman, 1967; Rittel and Webber 1973) describes certain types of problems as "wicked problems". For such wicked problems, the actual ‘problem’, the structure of the space, and the criteria for ‘evaluation’, only become clearly definable when the general form of the design ‘solution’ has been thrashed out. Such observations have led many researchers to create ever more elaborate search-based models of the design process. Some include multiple search spaces. Others describe complex organisations of spaces and sub-spaces. Some even incorporate dynamically evolving spaces that change over time. The necessity for such convoluted elaboration would seem to suggest that the search metaphor is not the most appropriate analogy. An alternative conceptualisation of the process of designing is required that does not 'go against the grain' of what intuitively designing seems to be about.

A description by J. K. Page, quoted by John Christopher Jones (1980), is suggested as the basis for developing such an alternative conceptualisation. Page describes designing as an ‘imaginative jump from present facts to future possibilities’ This metaphor captures a natural understanding of design in that it seems to start with a particular state and then suggests a reaction to this state, rather than starting with a generic space and suggesting a discovery of a pre-existing solution.

3.3 DESIGN SCHEMA

In *An Introduction to Evolutionary Design by Computers*, concerning evolutionary art systems Peter Bentley (1999) writes “One undesired side-effect of many of these representations is that they generate pieces of art which have very distinct styles”. Any generating process that creates design-proposals must inevitably employ various representations. Furthermore, the fact that these representations will have an affect on the type and style of the forms produced is also inevitable. However, referring to this as the 'style problem' and describing it as an 'undesired side effect' may not be the right approach. Instead, an alternative approach may be to exploit the 'style problem' to beneficial effect by restricting the scope of the generative system to a relatively specific subset or subcategory of designs. The question that remains is on what basis should this categorisation be made? From a designer's point of view, the most relevant category is the body of work that they have produced, are producing and will produce in the near future. The key aspect that unites this category is the particular designer's 'style'. This personal trademark has been described as a 'design-schema' (Janssen 2000, Frazer 2001). The design schema should not be interpreted as a superficial quality merely concerned with form and appearance of a design proposal. The design schema reflects a design stance and methodology and will often incorporates a complex conceptual framework and reasoning system.

The formative and the generative rules together encapsulate and encode the important concepts in a design schema. The formative defines a set of entities and relations, most of which will be part of the representational framework. However, some idiosyncratic concepts within the design schema are unlikely to be part of the representational framework. The formative may incorporate such concepts as additional entities and relations. This formative is a fundamental building block of an idiosyncratic generating process. The formative can be visualised as a densely overdesigned folly. There is no externalised function or purpose endorsing the formative. It is a tightly knit bundle that instantiates all the possible entities and relations of a particular design schema. The generative rules define how the entities and relations in the formative may be copied, deleted and modified.

3.4 THE ROAD MAP

The concepts of a formative with its associated generative rules are being further developed and implementation issues are being considered. From an implementation point of view, one promising and highly active area of research that has many similarities with these ideas is parametric design. Initially, parametric design referred to the use of parameters to define shape. Parametric design is here used in a much wider sense, to include a number of more recent developments. Javier Monedero (2000) presents a summary of the use of parametric design in architecture in which he identifies five types of parametric design: variants programming, history based constraint modellers, variational design, rule-based variants, and parametric feature-based design. A more recent addition, referred to as general case parametrics, is perhaps the most generalised approach to parametric design seems. Richard Wittenoom (1999) defines general case parametrics as "parametrics which includes but is not limited to geometrically founded parametrics and which may define abstract concepts or processes involving changes in model state and model topology".

Formatives and their associated generative rules overlap with many ideas found in highly generalised parametric systems. However, there are also a number of important differences. More research is required in order to discover how the overlap might be exploited.

4. Rational Foresight

4.1 PROCESS MODELS

In an article in World Architecture magazine, Colin Davis (2001) argues that as a force for change in our daily lives, architecture is growing weaker every year. "This estrangement from construction is the root cause of architecture's declining influence in society… Architecture's alliance with art - or what is usually today called 'art practice' - has become so inflated that it has eclipsed the much more essential alliance with construction". In essence, Davis is arguing that the architect should consider the process of construction rather than focus solely on the end product. Taking this argument one step further, the construction process may been seen as just one aspect of the holistic life cycle of a product. This life cycle is a continuous process that might include feasibility, design development, commissioning, manufacturing and building, deployment for intended function or purpose, gradual disuse and degeneration, and finally decommissioning and possibly even recycling. The most important characteristic of this outlook is that it is process orientated rather than product orientated. The product is seen as a time-extended and time-differentiated thing that is a part of the outcome.

Process models for the AEC/FM industry are now common. A number of standardisation efforts, including the International Standards Organisation (ISO) and the International Alliance on Interoperability (IAI), have proposed AEC/FM process models. Also, most national professional bodies use standard contractual stages for design services. There are many possible ways to partition the transformation of an initial design opportunity to final design recycling, each one exhibiting a particular bias. The G-P Soup is based on a generic process model that also exhibits a bias. In this case the bias emphasises a designers perspective. Three key transition points are identified: 'design opportunity', 'design proposal' and 'design outcome'. Two phases link these transition points. The first phase, referred to as the 'imagining-foreseeing' phase, starts with a design opportunity and develops this into a design proposal. The second phase, referred to as the 'materialisation-deployment' phase, starts with a design proposal and leads to a design outcome.

4.2 WHO TAKES RESPONSIBILITY?

The proposed model is not conceived as the 'correct' model. It is proposed as a device to highlight a certain point of view of the AEC/FM industry. Indeed, it is almost certainly not entirely accurate. The main characteristic of this point of view is the differentiation of two separate communities with distinct levels of responsibility. In the materialisation-deployment community, the scope of their responsibility is low. People perform tasks and they have a responsibility for performing these tasks properly. If the knock on effects of their task are some how undesirable, then it is not really their problem. Thus, the contractor will follow the specifications and drawings when constructing the building. If the building subsequently causes the inhabitants to become severely depressed, then the contractor is not to blame. In the imagining-foreseeing community, the scope of responsibility is high. People imagine desirable possibilities and try to foresee what changes may be introduced in order to allow such possibilities to unfold. Needless to say, they often get it wrong and the responsibility is theirs. The imagining-foreseeing community might have to accept a considerable part of the blame for the depressed inhabitants.

This model seems to promote a vision of the design proposal as a coherent set of documents, ceremoniously being handed over, from the imagining-foreseeing community to the materialisation-deployment community. In reality, this may be inaccurate in many ways. However, this vision emphasises the serious responsibilities that are embodied in the hand-over of the design proposal. Above, personal idiosyncrasy has been argued to be an essential ingredient in the process of creating design proposals. The responsibilities involved in creating design proposals highlight the other essential ingredient; rational foresight. Rational foresight allows the imagining-foreseeing community to make accurate guesses as to what the consequences in certain situations might be. These include many pragmatic issues such as how the building is to be constructed, the structural stability of a building during earthquakes, and the environmental behaviour of the building during different seasons. But also, subjective criteria are considered, such as the aesthetic appeal, the social impact and the psychological affect of the building.

4.3 SOFTWARE APPLICATIONS FOR MAKING PREDICTIONS

The G-P Soup aims to directly enhance the first phase in the process model; the imagining-foreseeing phase. The generating process enhances imagination and the predicting process enhances foresight. In particular, the predicting process uses various types of simulation and analysis software in order to make accurate predictions concerning certain aspects of the materialisation-deployment phase. These predictions may lead to the sinking of that particular design proposal.

An important aspect of the predicting process is the use of external analysis and simulation applications. Within the AEC/FM industry, many specialised types of simulation and analysis applications already exist and many more are now under development. For a computational design support environment to attempt to create all these applications 'in house' would not be feasible. Furthermore, it would be unlikely that such an environment would be able to keep up with technological developments in the simulation and analysis field.

A more desirable scenario would be the ability to 'plug-in' external analysis applications as and when required. In such a case, design proposals would need to be generated in a suitable format. The representational framework used is a key element in creating a suitable format. This representational framework must encompass all the concepts that the external analysis and simulation software applications are likely to use. For instance, Ardeshir Mahdavi (1998) identifies three types of simulation and analysis software as "a) component-based (e.g. cost estimation), b) space-based (e.g. thermal simulation), and c) network-based (e.g. structural analysis)". In order to allow all three types of application to 'plug-in' to the G-P Soup, the representational framework used will need to combine component, space and network based representations. In essence, this is a problem of knowledge representation.

4.4 THE ROAD MAP

Computer languages for the representation of knowledge have a long history. The emergence of object orientated languages over the last thirty years has led to new possibilities in the representation of knowledge. Objects can be thought of as modular pieces of code that encapsulates both state and behaviour. Thus an object instance that represents a physical entity such as a wall must contain both the data that describes the wall and the operators that allow one to create, manipulate and delete that wall. In the AEC/FM industry, the current generation of advanced applications represents the information that they are manipulating as objects with state and behaviour.

The International Alliance on Interoperability (IAI) has been working on such a model for the AEC/FM domain that defines and manages a set of entities, relations and constraints. The key deliverable of the IAI is the IFC Object Model that provides a formal specification of the requirements that can be used by the software authors. The IFC model incorporates many ideas from earlier modelling efforts. The IFC 2.x model encompasses a wide range of the concepts in use in the AEC/FM domain. These include project, site, building, building storey, space, building element, wall, door, column, beam, floor, roof, curtain wall, railing, covering, compartment, grid, nesting, assembly, connection, containment, cost, and so forth.

The accumulated research that has gone into the UFC model is very significant. Furthermore, the IFC model is supported by a number of key software developers in the AEC/FM industry, including Autodesk, Bentley, Graphisoft, and Nemetschek. Such industry commitment indicates that integration will become feasible in the near future. More research is required in order to understand how the IFC model may be utilised as a basis for implementing the representational framework.

5. Conclusions

"Our job is to give the architect *not* what he wants, but what he never dreamed he wanted; and when he gets it, he recognizes it as something that he wanted all the time" (adapted from Denys Lasdun 1965).

This paper has argued that the process of design does not aim to solve problems, but instead aims to create new possibilities. Similarly, this paper describes a G-P Soup that does not aim to solve some ‘problem’ with the existing design methodology. Instead, such an environment is one aspect of a new possibility. More specifically, this new possibility does not attempt to support or emulate the existing design methodology. Rather, it relies on a modified type of design methodology in tandem with a new type of computational environment.

This modified methodology, when applied within the G-P Soup environment, is expected to lead to design proposals that could not have arisen by any other means. The techniques used during the designing process and the design proposals created are therefore not seen to be independent from each other. This should come as no surprise. In "Translations from Drawing to Building", Robin Evans (1986) discusses another technique; parallel projection:

> Happy results do not of course occur under guarantee of the drawing technique, also requiring, as they do, an inquisitive mind, a very strong presentiment of the sense within forms, together with a penetrating ability to visualize spatial relations. This ability was doubtless enhanced by the practice of projective geometry, but not purchased with it. Still, it would be as crude to insist on the architect's unfettered imagination as the true source of forms, as it would to portray the drawing technique alone as the fount of formal invention. The point is that the imagination and the technique worked well together, the one enlarging the other, and the forms in question… could not have arisen other than through projection.

Similarly, the G-P Soup does not guarantee happy results. Instead, it enhances the ability of the designer to imagine and foresee.

6. Acknowledgements

Our research project is supported by a UGC PhD project grant from the Hong Kong Polytechnic University.

7. References

Aish, R.: 2000, Custom objects: A model oriented end-user programming environment, "*Visual End User" Workshop VL2000*.

Bentley, P.: 1999, An introduction to evolutionary design by computers, i*n* Bentley, P. (ed.), *Evolutionary Design by Computers,* Morgan. Kaufmann Publishers Inc, San Fransisco, pp. 1-73.

Churchman, C.: 1967, Wicked problems*, Management Science*, **14**(4), B141-142.

Cross, N.: 1999, Natural intelligence in design, *Design Studies,* **20**(1), pp 25-39.

Davies, C.: 2001, Who needs architecture?, *World Architecture*, **93**, 38-39.

Davis, H.: 1999, *The Culture of Building*, Oxford University Press, New York.

Dawkins, R.: 1983, Universal Darwinism, *in* D. S. Bendall (ed.), *Evolution from Molecules to Men,* Cambridge University Press, Cambridge, MA.

Eastman, C.: 1999, *Building Product Models: Computer Environments Supporting Design and Construction*, CRC Press, Boca Raton, Florida, USA.

Evans, R. (1986) Translations from drawing to building, *Translations From Drawing to Building and Other Essays,* Architectural Association Publications, London, pp. 153-193.

Frazer, J.: 2001, Creative design and the generative evolutionary paradigm. *in* P. J. Bentley and D. W. Corne (eds), *Creative Evolutionary Systems*, Morgan Kaufmann Pub, (to appear).

Humphrey, N.: 1987, Scientific Shakespeare, *in The Guardian,* London, August 26.

Janssen, P., Frazer, J. and Tang, M.X.: 2000, Evolutionary design systems: a conceptual framework for the creation of generative processes, *Proceedings of the 5th International Conference on Design Decision Support Systems in Architecture and Urban Planning*, Nijkerk, The Netherlands, pp. 190-200.

Janssen, P., Frazer, J. and Tang, M.X.: 2001, Evolutionary design systems and generative processes, *in* P. J. Bentley and D. W. Corne (eds), Special Issue on Creative Evolutionary Systems, *The International Journal of Artificial Intelligence, Neural Networks, and Complex Problem-Solving Technologies,* (to appear).

Jones, J. C.: 1980, *Design Methods,* Wiley, Chichester, UK.

Lasdun, D.: 1965, An architect's approach to architecture, *RIBA Journal,* **72**(4).

Mahdavi, A.: 1998, A Middle way to integration, *Proceedings of the 4th International Conference on Design Decision Support Systems in Architecture and Urban Planning*, Maastricht, The Netherlands.

Monedero, J.: 2000, Parametric design: A review and some experiences, *Automation in Construction*, **9**(4), 369-377.

Newell, A., Shaw, J. C. and Simon, H. A.: 1957, The process of creative thinking, *in* H. Gruber, G. Terrell, and M. Wertheimer, (eds), *Contemporary Approaches to Creative Thinking*, Atherton Press, New York, pp. 63-119.

Rittel, H. W. J. and Webber, M. M.: 1973, Dilemmas in a general theory of planning, *Policy Sciences,* **4**, 155-169.

Wittenoom, R.: 1999, General case parametric representation and exchange - Future directions, *ISO/WD Technical Report Type 3*, ISO TC184/SC4/WG12 N292, National Institute of Standards Technology, Gaithersburg, Maryland, USA.