Integrating Generative Modelling and Design Process

An Interface for Visual Design

TALBOTT Kyle

Milwaukee School of Architecture and Urban Planning, University of Wisconsin, USA

Keywords: human-computer interaction, generative design, creativity, design process, 3D modelling

Abstract: While researchers learn to generate design options with a computer, the question of how a designer interacts with these options remains largely unexcavated. This paper explores how a generative system should interact with a designer. It argues that limitations in human cognition require a designer to manage the complexity of his task with a creative thinking strategy called Perception-Imagination Overlay. To support this, a generative system should present a progression of contextual design "moves" rather than a collection of finished solutions. After validating a set of interface requirements, the paper presents an example interface. The paper concludes that the development of computational methods cannot be legitimately isolated from the development of an appropriate designer-system interaction, and that integrating these concerns might transform our conception of generative systems.

1 INTRODUCTION: INTERACTING WITH A GENERATIVE MODELING SYSTEM

To support the work of architectural designers, it is often held that a generative modelling system needs to produce creative design options. A creative option is 1) not known beforehand, 2) relevant to the design task at hand (Schmidt-Belz and Hovestadt 1996), and 3) not found by rote application of formula (Janssen, Frazer and Tang 2000). To the extent it is successful in this endeavour, a generative system externalizes an aspect of creative thinking. The creative effort is divided – part conducted by a designer and part conducted by the system. In order for this dual effort to produce a coherent result, designer and system must at some point interact and exchange the product of their labour.

The degree of designer-system interaction supported by current prototype systems is generally low (Heylighen and Neuckermanns 2003). Oxman and Heylighen (2001) have recognized the lack of graphical content, arguing that because visual thinking is fundamental to design, this factor should be addressed from the outset. A weak interest in interaction can be seen in researchers' descriptions of prototype systems. With few exceptions, they offer thin information about interface, and instead describe computational methods for generating options. A diversity of computational methods results, including case-based reasoning, rule-based reasoning, genetic algorithms and neural networks – each rich in possibilities. However, most systems support a similar sort of designer-system interaction.

The paper examines this popular approach to interaction and demonstrates an incompatibility with designer needs. It then proposes an alternative approach that avoids the identified problem. The alternative is concretized in a fictional interface, which is described through a series of interaction scenarios.

The goal of this research is not to construct a prototype system, but to establish a method of interaction appropriate to such systems generally, which could help software engineers develop a practical design tool incorporating generative modelling capabilities. The interface described here shows only one possible manifestation of the suggested method of interaction. It is not offered as a definitive solution, but as a demonstration of principles and as an idea generator.

2 CREATIVE THINKING IN CONVENTIONAL MODEL-BUILDING

The criticism levied here against the usual approach is rooted in a specific method of creative thinking. It occurs as a designer constructs, observes and manipulates a study model. In order to generate new design options with this medium, he must engage the model-building process as a creative exploration and not merely as rote documentation of a preconceived product. In this frame of mind, new possibilities can arise from a variety of sources, such as the happy accident of model construction that improves a design (Goldschmidt 1994), limitations in properties of a model-building material that reveal new methods of construction (Moloney and Issa 2003), or the recognition of unintended consequences on one domain while attempting to improve another (Schon and Wiggins 1992).

The routes to creative insight mentioned above do not result from an intentional act of option generation. Then come from the recognition of opportunities in an observed circumstance. While such methods are valuable, they do not provide a complete picture of designer creativity. With only these means, a designer is put in the position of passive spectator, waiting for an answer to reveal itself. Yet we observe a more deliberate manner in the efforts of a skilled designer. He actively pursues a goal. He searches in a directed manner and makes discoveries that do not seem fully explained by circumstance.

A designer can also create by *transforming a study model in imagination*. Using this method, he does not need to wait for a study model to reveal new options; he actively generates new options. Because imagination is limited in the amount of complexity it can handle, it cannot stand alone. Instead, imagined content “fills in the gaps” in a study model. The result is a composite form of mental imagery, generated partly by a perceived study model and partly by an imagined extension from it.

That humans possess this power of *Perception-Imagination Overlay (PIO)* is indicated by an example from common experience. One can look out into a room and, with one's eyes open, imagine a person walking through it. The person is imaginary and the room is real, yet the result is a single, composite mental image. Depending on one's capacity to imagine, the person might appear dim and indistinct compared to the surrounding room, but it is overlaid there nonetheless.

In design, this ability is used to generate and test alternatives. For instance, a roof plane can be extended or rotated in imagination, the position of a column can be shifted, or the number of mullions in a curtain wall can be multiplied. Overlaying imagined alternatives on a perceived study model can occur continuously as a designer builds.

A designer benefits from this power. Although the amount of information he can hold in imagination is limited, perception is not so limited. We perceive a field of complexity effortlessly. However, the content of perception cannot be wilfully altered in the mind – a feat afforded only to imagination. By capturing much of a building's complexity in a study model, which he perceives, a designer allocates to imagination only those elements to be mentally transformed. In this way the special power of each mode of visual imagery is unified in a single mental act.

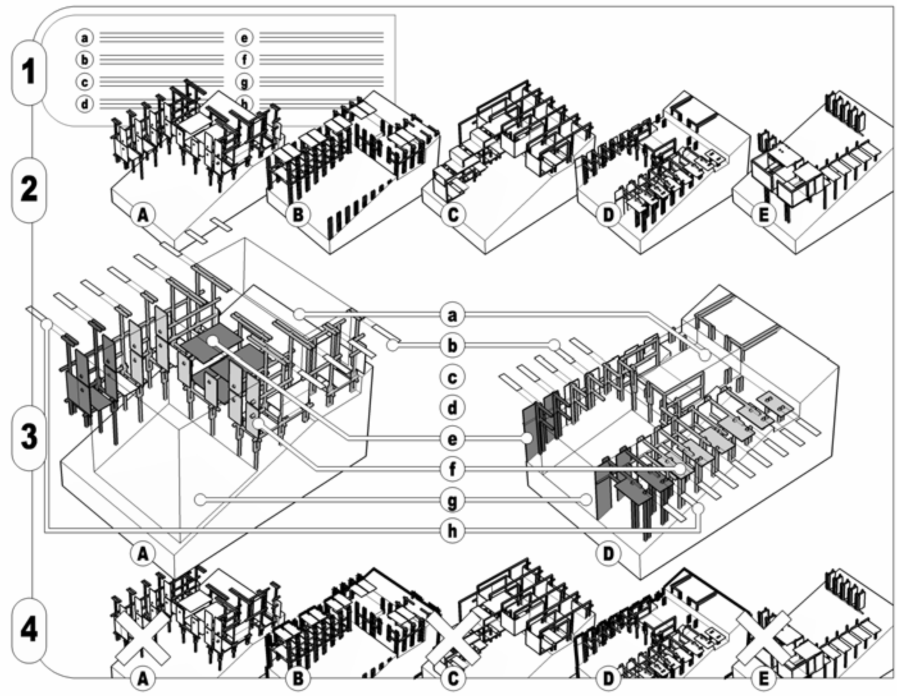

Figure 1 A Popular Approach to Designer-System Interaction

This method of creative thinking provides a useful perspective from which to critique the currently popular idea of how a designer should interact with a generative modelling system.

3 DESCRIPTION OF A POPULAR APPROACH TO DESIGNER-SYSTEM INTERACTION

From the point of view of a designer, generative modelling systems work in basically the same way despite different technologies “under the hood.” Examples include Garza and Maher (2001), Elezkurtaj and Frank (2000), and Silva and Bridges (1997). In each case a designer is first asked to input precedents or performance criteria to constrain the search (Figure 1.1). The system begins its work, and he is confronted with a series or collection of system-generated whole solutions (Figure 1.2), which he is asked to evaluate (Figure 1.3). He selects one or more of these as preferable (Figure 1.4), and then feeds them back into the system to seed further option generation. The system then presents another batch for review, and the process cycles until an acceptable design is found.

4 CRITIQUE OF THE POPULAR APPROACH

When constructing a study model, each intermediary model state is one step in an accumulating progression of design “moves” (Arlati, Bottelli and Fogh 1996). As Schon and Wiggins (1992) have described, this progression helps manage the mental effort exerted in creative thinking. The full complexity of a building is broken down and evaluated piecemeal. In contrast, a designer using a typical generative system sees only the end product of the system’s calculations at an already advanced state of solution (Arlati, Bottelli and Fogh 1996). What the system offers is not a single design “move” but a complex bundle of “moves” presented at once. This difference between conventional and generative modelling is problematic.

As construction progresses, a study model allows new “moves” to be compared directly against the model without consulting the design goals. As a designer builds, the basic question shifts from: “Does this ‘move’ satisfy my design goals?” to: “Does this ‘move’ fit into and reinforce the rest of my composition (which I already know satisfies my design goals)?” This later question can be answered by a lightning fast perceptual assessment rather than a time-consuming goal analysis. A study model summarizes all the piecemeal evaluating a designer did to reach it, and therefore, it is a material embodiment of the design goals made accessible directly to perception.

PIO takes advantage of this by combining two processes – the generation and evaluation of design “moves.” The overlay of new “moves” in imagination and the weighing of these “moves” against an existing study model are experienced as a seamless act, in which knowing the rightness or wrongness of the “move” is immediate and self-evident. It does not feel like evaluation is being conducted; it happens effortlessly.

The popular approach to designer-system interaction undermines the thought process described above by eliminating a sense of progression in the viewing of design "moves". Each option generated by the system is not only a complex bundle of "moves," it is a *freestanding* bundle without any necessary similarity to the one generated before or after. Each one must be confronted as a unique entity, requiring a designer to repeat the arduous process of evaluating it in full.

5 A PROPOSED INTERFACE

Proposed here is a fictional interface to a generative modelling system based on a different approach to designer-system interaction – one that emphasizes the presentation of *progressive partial solutions rather than freestanding whole solutions*. The interface tackles a limited scope – supporting the tectonic composition of solids in space. Although the examples show simple situations in which orthogonal objects are composed additively, such an interface could support the subtraction of volumes, the iterative alteration of existing geometry and design at other scales.

The interface overcomes deficiencies identified in the preceding critique by respecting the following designer needs related to PIO.

- A designer needs to view system-generated options in partial states of completion.
- A designer needs to evaluate system-generated options progressively based on perceptual criteria applied in a visual context. (He should not be forced to regress to a list of abstract goals or pre-defined performance criteria, which shifts him out of a visual mode of thinking.)
- System-generated options need to occur locally, unfolding at the point in a study model where a designer currently conducts PIO.
- System-generated options need to assist a designer in addressing whatever issues he currently contemplates, and not force him to shift his train of thought to other issues. (This is needed to produce a sense of progression in design "moves".)

Interaction with such an interface might happen as follows. A designer constructs digital objects and begins accumulating them in a study model, much like conventional modelling. He then constructs an object that in his mind represents a column, for instance, and positions it where he thinks it might go (Figure 2.1, light grey). Because he is tentative in his conviction and wants to consider other possibilities, he prompts the system (by pressing the Shift key) to show him other things to do with the column. The system begins showing him alternative locations for the column, allowing him to scroll through a series of options (Figure 2.1a-c, dark grey). With each click, a ghosted object(s) is superimposed on the model, showing the visual result of the system's calculations. If he sees a desirable option, he accepts it with a click of the mouse (Figure 2.1d), which prompts the system to alter the model accordingly (Figure 2.2). He then continues building the model, adding and modifying pieces in the conventional manner.

This process might seem conservative compared to a self-running generative system, since it treats automated option generation as a sub-routine within an otherwise conventional model-building process. However, the same approach could be used with varying aggressiveness. A seasoned designer might make an occasional request to the system when his imagination stalls, while a novice might use it frequently. An example of the later is mapped in Figure 2.1-2.3. Here, a designer engages the system in a sustained dialogue, using it to generate alternatives at every step of model construction. A further degree of control is given over to the system in this case, using output selected at the end of step two (Figure 2.2d), for instance, as input for a new round of automated option generation in step three.

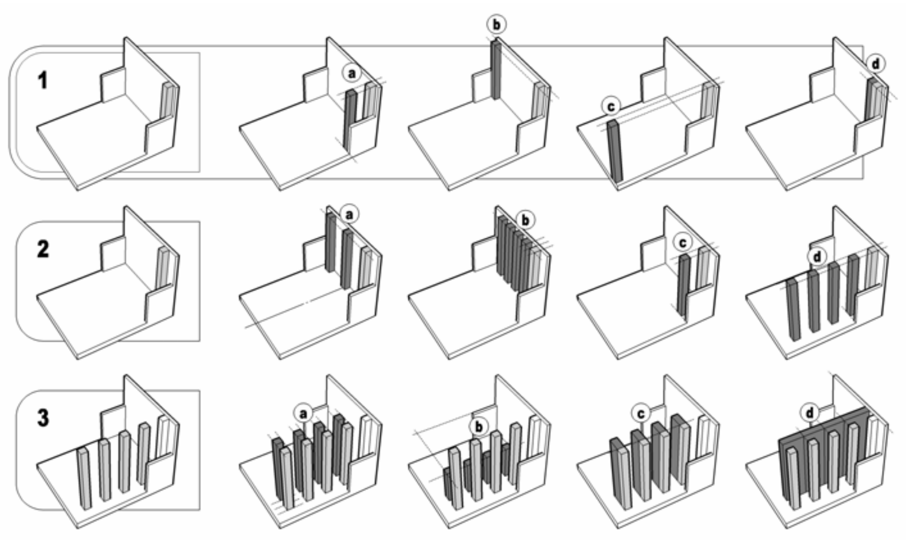

Figure 2 Design Scenarios using the Proposed Interface

These simple examples demonstrate some features of the interface. First, options are generated in reference to the current partial model, accounting for both a target object(s) (currently of interest to a designer) as well as objects in the vicinity. This localizing of the generative calculation reduces the number of variables involved, shifting the question away from: "How can we generate comprehensive building solutions?" (Schmidt-Belz and Hovestadt 1996) – to: "How can we recognize and track relationships between objects and make a context-sensitive suggestion?" Instead of producing a comprehensive design, localization allows simple options to accumulate progressively into a complex whole, in a manner consistent with design thinking.

Second, a designer's actions interlace with the actions of the system. Rather than sit back and let the computer produce a batch of options independently, here he frequently coaxes the system. This active guidance increases the likelihood that system-generated options are relevant to issues actively contemplated (Lund 2000). This maintains an additional parallel between thought process and computational process. And further, the interface offers at least four degrees of control from 1) complete designer control to 2) occasional advice from the system to 3) sustained advice from the system to 4) system-generated modelling with occasional designer intervention.

Finally, the system shows options graphically in the context of surrounding, already evaluated objects. This allows a designer to judge the appropriateness of system-generated options by the standard of visual integration.

Having examined a sample designer-system interaction, we can now dissect the interface in more detail.

6 GENERATING RELEVANT OPTIONS IN THE PROPOSED INTERFACE

System-generated options cannot be random, or many irrelevant options might be viewed before finding something useful (Janssen, Frazer and Tang 2000, Soufi and Edmonds 1996). To filter irrelevant options, systems often rely on persistent and explicit evaluative rules, which are not easily input or adjusted by a designer (Arlati, Bottelli and Fogh 1996, Goldschmidt, 1994). While this approach might be adequate for some tasks – those narrowly defined and sufficiently quantifiable – it can not generally satisfy a designer's needs (Elezkurtaj and Franck 2000). Design goals are revised, deleted and spawned intermittently, applied to shifting contexts and often left unstated (Goldschmidt 1997). To be compatible with designer needs, a system must filter options while respecting the fluid and tacit nature of design goals.

The method of PIO in conventional modelling provides a paradigm for relevancy filtering that meets this challenge.

When a designer uses PIO, options are filtered in two ways. First, the imagined portion of the overlay is constrained by a limited means of mental transformation. Three transformations – Reposition, Extension, and Distortion – seem to be inherent in the capacity to imagine, and they explain many imagined alternatives (Talbott 2004). Second, imagination is constrained by the perceived portion of the overlay. Without some degree of visual correspondence between imagination and study model, the overlay would be meaningless.

These constraints are inherent in the method of PIO and independent of a designer's specific goals. They contribute to a universal method of visual search used to generate and evaluate options according to their consistency with a body of already evaluated design "moves".

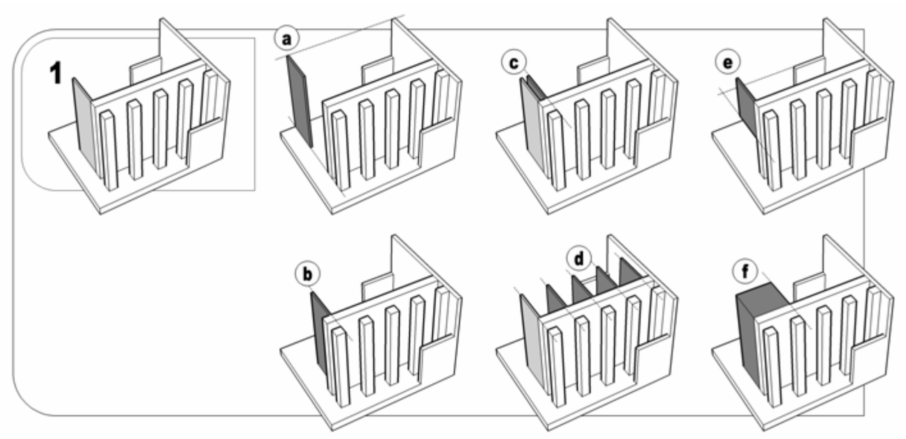

Figure 3 Transformations: Repositioning, Extending and Distorting

An automated version of this process guides the proposed interface. An option is generated by applying one of three transformations to a target object(s), and the result is constrained by visual correspondence with surrounding objects. The target object possesses both a start shape and position (Figure 3.1, light grey). *Repositioning* changes the position of the object (Figure 3.a-b, dark grey), *Extending* duplicates the object in a repetitive series (Figure 3.c-d), and *Distorting* changes the shape of the object (Figure 3.e-f). Each generated option is then tested for relevancy with three criteria of visual correspondence: 1) the generated object occurs in the immediate vicinity as determined by the scope of objects displayed in the active window, 2) the generated object inhabits its own space as determined by an interference check with vicinity objects, and 3) the generated object exhibits a minimum number of alignments with vicinity objects. Additionally, each alignment is weighted according to the start conditions of the target object. Direct alignment with the start conditions receives heavier weighting, as do alignments with vicinity objects closer to the start conditions (Figure 4).

This set of rules is capable of generating a wide range of options that remain compositionally relevant (through proximity and alignment) to a designer-defined context. It is important to note that a designer controls not only the scope of the context (by positioning the active view window), but also the degree to which system-generated options adhere to or diverge from the context. With a simple set of sliders in a dialogue box, he balances three contributing factors: “Personalized,” “Aligned” and “Transformed.”

The first slider controls the degree of customized option generation. A designer’s history of choices among generated options is recorded, eventually establishing preferences for type of transformation and type and weighting of alignments. Priority can then be given to favourite types by pushing the slider toward the “+” end of the “Personalized” item. If a designer wants to diversify the search, he can push the slider to the “-” end, in which case the system discounts his history. The second slider controls the number of alignments needed to form a relevant option.

By pushing the slider toward the "+" end of the "Aligned" item, a closer similarity is maintained between generated options and vicinity objects. As the slider moves toward the "-" end, generated options deviate increasingly in shape and position from their surroundings. The third slider controls the number of transformations performed. As the slider is pushed toward "+" the number of repositioning, extending and distorting operations used to generate each option is increased, while a slider positioned at "-" applies only one transformation. Figure 5.a-f shows a series of options generated with the "Transformed" slider moved toward "+." Here each option is the result of a combination of three transformations. In comparison to the target object (the beam shown at upper left), the generated options (dark grey) suggest more divergent possibilities.

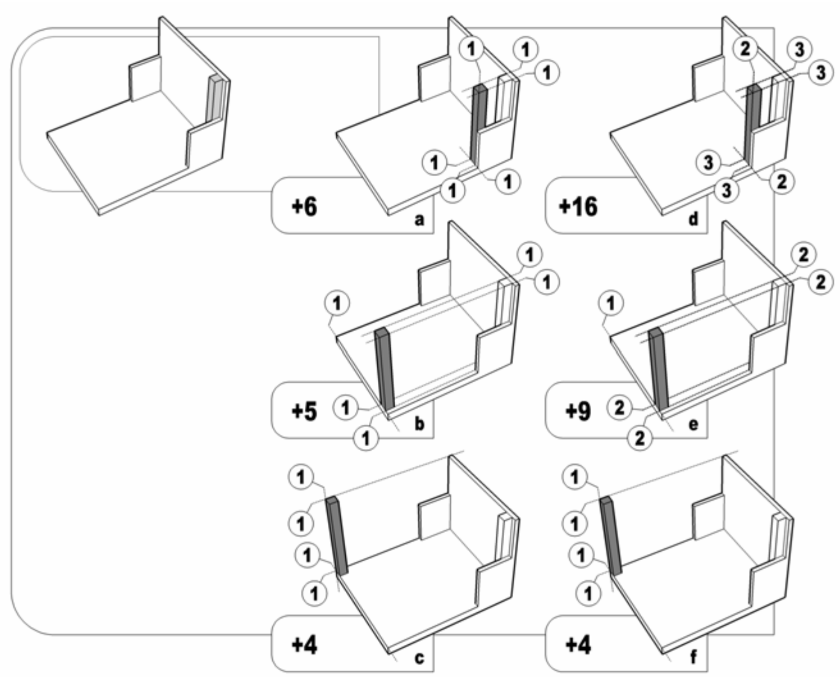

Figure 4 Alignments (Left Column) and Alignment Weighting (Right Column)

When an option passes the tests of relevancy, the system displays it as a semi-transparent overlay on the study model. It is then evaluated by a designer as reinforcing or undermining the existing composition of elements. No effort is made by the system to make this evaluation, nor to assess fitness to any purpose. A designer may hold his goals in any form desired, evaluate options in any way desired, and change his goals at any point in the process without the need to freeze them, quantify them, and inform the system. In this way the interface maintains a distinction between universal filters, which are determined by cognition and context, and variable filters, which are determined by designer choice. By automating only universal filters, the interface leaves design goals free to evolve.

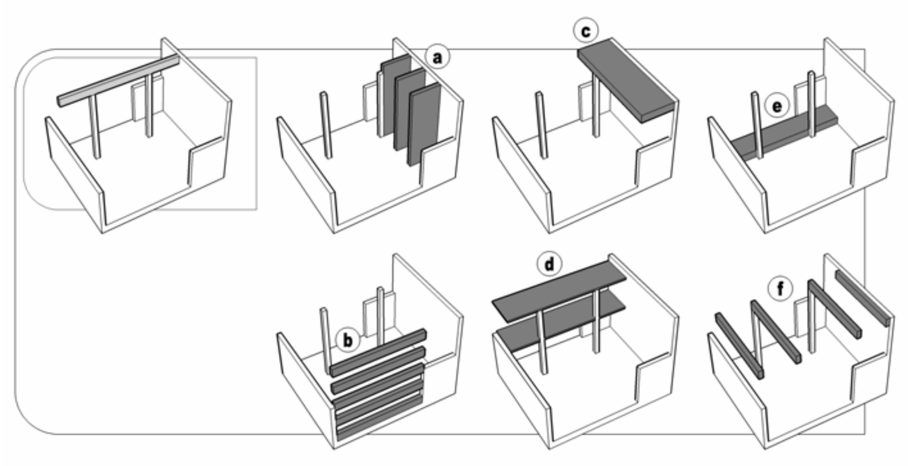

Figure 5 System-Generated Options with Higher Degree of Transformation

Two additional points should be made here about creativity, which indicate the degree to which the interface departs from the usual approach.

First, instead of working from the premise that a designer’s creative effort is driven by quantifiable goal, architectural precedent, or a combination, the interface acknowledges two other driving forces: a designer’s imagination and the need to respond to a context of existing design “moves.” The proposed interface, therefore, emphasizes a different kind of designing – moving away from the analytical methods dominant in most generative modelling systems.

Second, a designer’s use of PIO can continue with the proposed interface. The generative system merely adds an interstitial layer of visual content – 1) existing model objects (Figure 6: white), 2) system-generated transformations (Figure 6: light grey), and 3) designer imagined transformations of *those* transformations (Figure 6: dark grey). In this way system-generated options are a mid-stream prompt to extend a designer’s use of PIO. Instead of introducing a different design method without clear relationship to a designer’s cognitive needs, here the design method is a modification of a traditional one that already accommodates those needs. PIO becomes PSOIO: Perception > System-generated Overlay > Imagination Overlay.

The central value of the proposed interface, then, is to extend the range of options explored using PIO by introducing a computer-generated intermediary. The same PIO conducted in conventional modelling is magnified to increase opportunities for creative discovery. The interface could also have pedagogical value. Since PIO requires imagination, and since imagination is a skill held by designers in various degrees, the ability to externalize the “What if?” propositions of PIO could help young designers learn to apply this important creative thinking strategy.

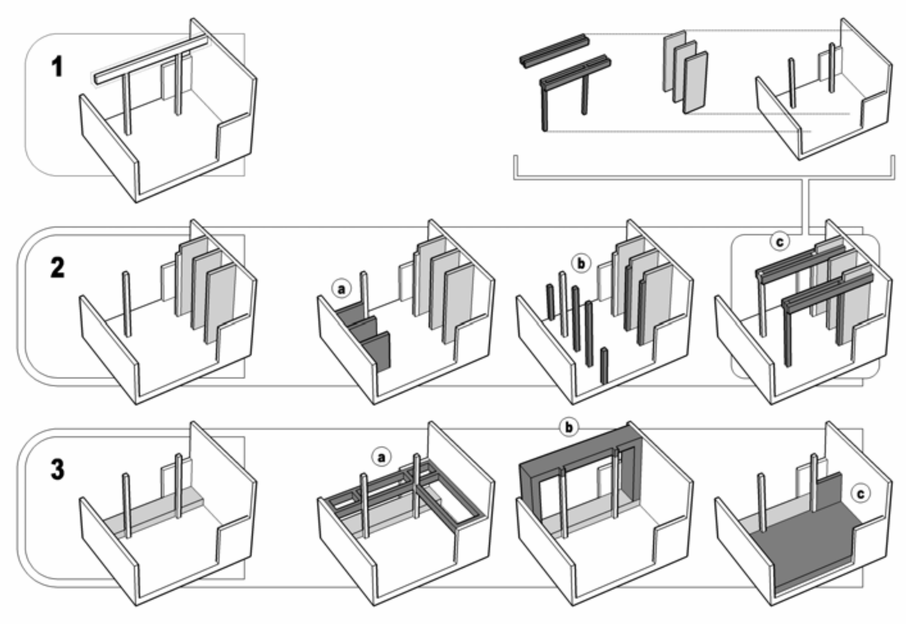

Figure 6 System-Augmented PIO

7 CONCLUSION

This paper examines a specific creative thinking method – Perception-Imagination Overlay – and explores its relationship to generative modelling. The goal was to identify a new point of integration between generative modelling systems and the architectural design process, and to develop an alternative approach to designer-system interaction in such systems.

REFERENCES

Arlati, E., V. Bottelli, and C. Fogh. 1996. Applying CBR to the Teaching of Architectural Design. *Education for Practice – 14th eCAADe Conference Proceedings*, Lund, Sweden, 41-50.

Elezkurtaj, T., and G. Franck. 2000. Geometry and Topology: A User-Interface to Artificial Evolution in Architectural Design. *Promise and Reality: State of the Art versus State of Practice in Computing for the Design and Planning Process – 18th eCAADe Conference Proceedings* 309-312.

Garza, A.G.S., and M.U. Maher. 2001. Using Evolutionary Methods for Design Case Adaptation. *Reinventing the Discourse: Proceedings of the Twenty First Annual Conference of the Association for Computer-Aided Design in Architecture*, 180-191. Buffalo, New York.

Goldschmidt, G. 1997. Capturing Indeterminism: Representation in the Design Problem Space. *Design Studies* 18(4): 441-455.

Goldschmidt, G. 1994. On Visual Design Thinking: The Vis Kids of Architecture. *Design Studies* 15(2): 158-174.

Heylighen, A., and H. Neuckermans. 2003. (Learning from Experience)?: Promises, Problems and Side-effects of Case-Based Reasoning in Architectural Design. *International Journal of Architectural Computing* 1(1): 60-70.

Janssen, P., J. Frazer, and M. Tang. 2000. Evolutionary Design Systems: A Conceptual Framework for the Creation of Generative Processes. In *Proceedings of the 5th International Conference on Design Decision Support Systems in Architecture and Urban Planning*, 190-200. Nijkerk, The Netherlands.

Lund, A. 2000. Evolving the Shape of Things to Come – A Comparison of Interactive Evolution and Direct Manipulation for Creative Tasks. *International Conference on Generative Art*, 1-13.

Moloney, J., and R. Issa. 2003. Materials in Architectural Design Education Software: A Case Study. *International Journal of Architectural Computing* 1(1): 46-58.

Oxman, R., and A. Heylighen. 2001. A Case with a View: Towards an Integration of Visual and Case-Based Reasoning in Design. *Architectural Information Management – 19th eCAADe Conference Proceedings*, 336-341. Helsinki, Finland.

Schmidt-Belz, B., and L. Hovestadt. 1996. Scenario of an Integrated Design Support for Architects. *Design Studies* 17: 489-509.

Schon, D.A., and G. Wiggins. 1992. Kinds of Seeing and Their Functions in Designing. *Design Studies* 13: 135-156.

Silva, N.F., and A.H. Bridges. 1997. Human-Computer Interaction and Neural Networks in Architectural Design: A Tool for Design Exploration. *CAAD Futures 1997 Conference Proceedings*, 267-284.

Soufi, B., and E. Edmonds. 1996. The Cognitive Basis of Emergence: Implications for Design Support. *Design Studies* 17: 451-463.

Talbott, K. 2004. Divergent Thinking in the Construction of Architectural Models. *International Journal of Architectural Computing* 2(2): 263-286.