DESIGN CONVERSATIONS WITH YOUR COMPUTER: EVALUATING EXPERIENTIAL QUALITIES OF PHYSICAL FORM

KIMBERLE KOILE
MITArtificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, MA 02139 USA

Abstract. This paper describes a prototype system that evaluates an architectural design using the designers theories about how to manifest experiential qualities in physical form. The system uses AT methods in conjunction with geometric and non-geometric knowledge to represent experiential qualities, e.g. privacy, in terms of concrete details of a design, e.g. wall dimensions and locations. This paper describes the organization and implementation of the system, and reports the results of an experiment in which the system was used to evaluate Frank Lloyd Wright Prairie houses.

1. Introduction

Imagine that you're sketching a design, pen in hand. You tell the computer near you that the design is for a family of four who want a house that feels both spacious and cozy; that invites the community to visit, but protects their privacy; that is just large enough for their needs, but not larger. You have ideas about circulation patterns and about physical forms that manifest feelings of privacy, and you're translating those ideas into lines and annotations on a page. You stop to assess your latest sketch and ask the computer for its comments. It graphically shows you access from exterior to interior and within the interior. It shows you regions defined by your proposed physical forms. It tells you that the main living space is not very private with respect to the exterior, but that it is spacious. It suggests moving the front door to increase the privacy. It suggests adding an entry territory between the front door and the main living space, and points out that if the area of the main living space decreases as a result, then the spaciousness may decrease. You like the first suggestion and accept it, but not the second. You continue sketching and evaluating.

This paper describes a prototype system capable of an important part of the above scenario: evaluation of an architectural design using the designer's theories and preferences about how to manifest experiential qualities in physical form.

2. Overview

The evaluation system described here is the first component of a prototype design support system for architectural design. The design support system adopts the prevalent view that architectural design is an iterative process of design generation, evaluation, and modification. The system assumes that a designer will generate a design, e.g. via sketching, then will evaluate and modify that design iteratively, asking the system: Does this design adhere to a set of specified design goals? If not, how can it be modified so that it does? Such a design support system will engage a designer in a conversation of the sort in the opening scenario

The design support system under development differs from other work in computer-aided architectural design in two ways:

1. The system focuses on architectural knowledge not often found in design systems: what architects know about manifesting experiential qualities in physical form. While some architects may design with only form in mind, most create spaces that people inhabit. They and their clients describe spaces as private, sunny, open, spacious, etc. Architects use their knowledge from past experiences, from environment behavior research, and from their own theories to create spaces with experiential qualities such as these. This knowledge can be articulated and structured as general design principles (e.g. Alexander, et al., 1977; Zeisel and Welch, 1981; Hertzberger, 1993), which can serve as a basis for a design support system that reasons about experiential qualities and physical form as in the opening scenario.

2. The system bridges the gap between knowledge-based systems and CAD systems by using AT methods in conjunction with both geometric and non-geometric knowledge. Al methods are ideal in domains rich with complex, subjective, heuristic knowledge, and design has been an important focus of Al research for many years (e.g. Coyne, et al., 1990; Pham, 1991; Gero and Sudweeks, 1996). Much of that research, especially in the domain of architecture, has focused on knowledge-based or case-based systems, with emphasis on non-geometric knowledge, such as topological relationships. Geometric knowledge has generally been left to CAD drawing systems. The use of Al methods in conjunction with both geometric and non-geometric knowledge will enable a tool to be built such as the one described in the opening scenario, which reasons about both experiential qualities and concrete details of physical form.

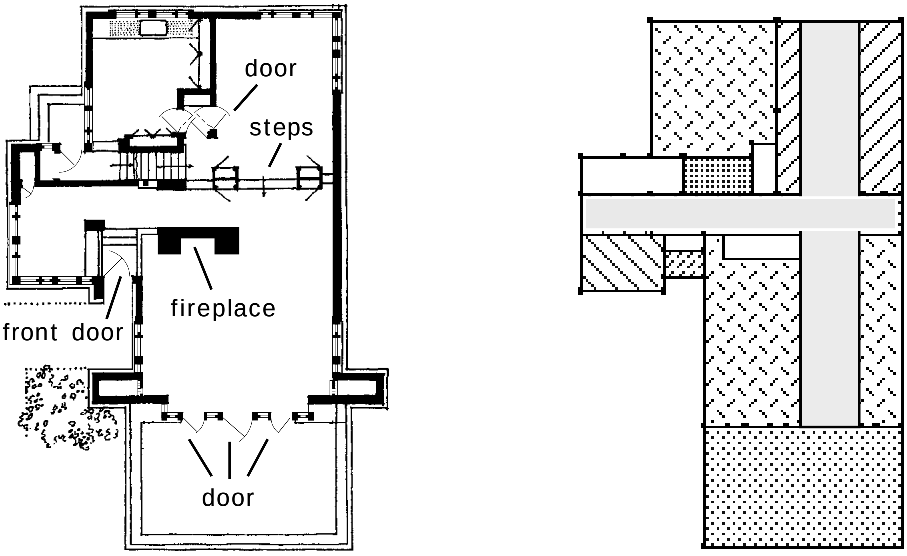
The design support system under development will be a step toward a design tool that acts as an intelligent design assistant. In particular, it will aid a designer by easily and quickly generating and keeping track of alternatives, by managing the complexity of many conflicting design goals, by providing more systematic exploration of a design space, and by serving as a repository for reusable design knowledge. The system also will contribute to the clarification of terms used in architectural discourse.

3. Evaluation System

The evaluation component of the design support system described in the previous section assesses architectural designs for the presence or absence of experiential qualities. For example: Is the main living space private? The evaluation system is organized around two ideas: that experiential qualities are manifested in physical form by means of the concrete details of a design; and that the knowledge relating qualities to details can be explicitly represented and manipulated in a reasoning system. The system, thus, represents and reasons about design details, experiential qualities, and the relationships between them.

3.1 DESIGN MODELS

A design is represented by several kinds of design models-a design-element model, a circulation model, a territory model, a use-space model, and a connectivity model. The design-element model contains information about design elements, which represent physical objects used to create form, e.g. walls, windows, doorways. The circulation model is a geometric abstraction of the design-element model, represented as a graph with nodes for doorways and links between doorways. The nodes and links retain dimension and location information. The territory model is another geometric abstraction of the design-element model; it contains lists of points, lines, and territories-regions of space derived from the design elements. The use-space model contains a list of use-spaces, which are activity areas defined by the user by assigning activity labels to particular territories. Representing use-spaces and territories separately enables the system to evaluate the physical form itself without necessarily tying it to intended use, and vice versa. Finally, the connectivity model is a representation of topological relationships between use-spaces; it is a graph with nodes for use-spaces and links between use-spaces directly connected to each other (e.g. by means of a shared doorway). See Figures 1 and 2 for diagrams of design models for the Mrs. Thomas Gale house.

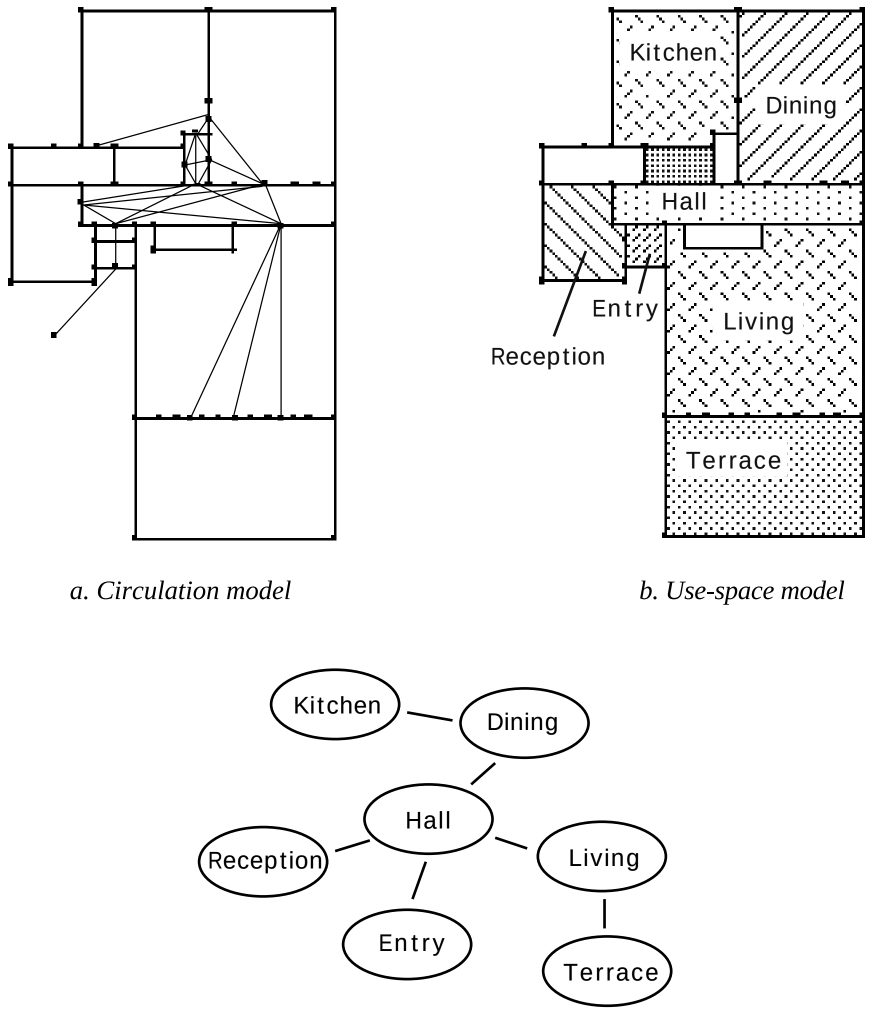


a. Design elements

b. Territory model

Figure 1. a. Floorplan[1] showing several design elements in the design-element model.
b. Territory model: territories formed by design elements, including two overlapping territories.

[1] (Storrer, 1993)



a. Circulation model

b. Use-space model

c. Connectivity model

Figure 2.a. Circulation model:[2] *paths from experior approach point through interior.*
b. Use-space model: use-spaces, which are territories plus activity labels.
c. Connectivity model. topological relationships between use-spaces.

[2] The circulation model is derived automatically from the design-element and territory models; the connectivity model from the use-space model. Design-element, territory, and use-space models currently are entered by hand. In future versions, design-element and use-space models will be derived automatically from a sketch (Gross, 1996); territory models will be derived automatically from the design-element model (Kincaid, 1997).

3.2 DESIGN CHARACTERISTICS

At the core of the system is a hierarchy[3] of design characteristics that provides mappings between experiential qualities and concrete architectural details that realize those qualities. The hierarchy represents knowledge such as: A place may feel private if it feels hidden. A place may feel hidden if it is reached via a circuitous path. It may also feel hidden if it is on a different level from other places, or if it is not entered directly from a place likely to be frequented by many people. In this example, feeling private is an experiential quality which can be achieved in a building by means of concrete architectural details, such as particular materials, locations and dimensions of walls, windows, etc. Both experiential qualities and concrete architectural details are represented as design characteristics.

Concrete details are referred to as primary design characteristics; they can be objectively measured. Experiential qualities are referred to as secondary design characteristics; they cannot be objectively measured; their values are based on primary design characteristics.

Both kinds of design characteristics are defined in terms of attributes of design objects[4] and relations between design objects, or as combinations of other design characteristics. Design characteristics can be combined via boolean connectives, arithmetic relations, arithmetic operators, or function composition.

The hierarchy described above provides mappings between secondary design characteristics and primary characteristics that realize them. A mapping between a secondary characteristic and primary characteristics need not be direct, but may pass through intermediate, more specific secondary characteristics. The most general secondary characteristics are near the top of the hierarchy, more specific ones closer to the bottom, with the bottommost level of secondary characteristics mapped directly to primary characteristics, which form the bottommost level of the hierarchy. See Figure 3 for an example showing a mapping from the secondary characteristic is-private to primary characteristics.

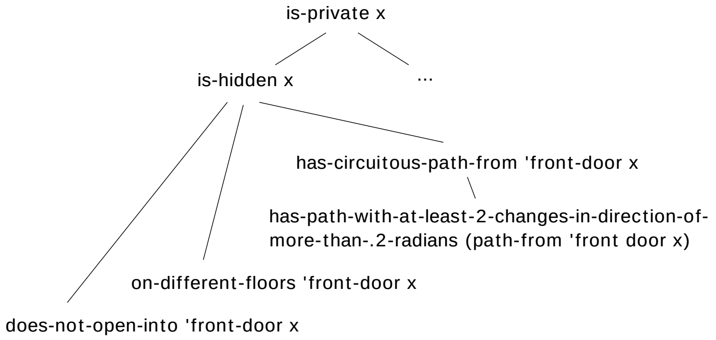

is-private x
is-hidden x
...
has-circuitous-path-from 'front-door x
has-path-with-at-least-2-changes-in-direction-of-more-than-.2-radians (path-from 'front door x)
on-different-floors 'front-door x
does-not-open-into 'front-door x

Figure 3. Portion of design characteristics hierarchy.

[3] It has multiple roots so is not a strict hierarchy. It was defined by the author with help from architect Richard Krauss, of Arrowstreet Inc.; future versions will be defined by system users.
[4] A design object is any design element, territory, use-space, or entire design.

3.3 EVALUATION

To evaluate a design, the user of the system specifies a design characteristic or list of design characteristics and any of the designs use-spaces, territories, or design elements. The system evaluates the design by traversing the design characteristics hierarchy to find and evaluate the appropriate primary design characteristics.

For example, given the design characteristic is-private and a territory, the system evaluates is-private for the territory by referring to the design characteristics hierarchy. It determines that is-private for a territory can be evaluated by looking for a circuitous path from the front door to the territory. A circuitous path is defined in terms of the circulation model: as a path containing at least two changes in direction of greater than .2 radians. So to evaluate is-private for a territory, the system evaluates the territory's associated circulationmodel for the presence of such a path from the front door.[5] The system also checks whether the front door opens into the territory and whether the territory and front door are on different levels.

3.4. IMPLEMENTATION

Design models and their associated components (e.g. design elements, territories, points) are implemented as Common Lisp Object System (CLOS) objects (Bobrow, et al., 1988). Object classes are arranged in a hierarchy in which each class can inherit from multiple parents; methods, specialized by class on any number of arguments, are used to dispatch generic functions on particular objects.

Design characteristics are represented as predicates in a rule-based, backward-chaining system.[6] Predicates for primary characteristics are each associated with an evaluator function, which operates on a design model or models. The value of a primary characteristic is the value returned by its evaluator function. All secondary design characteristics are boolean-valued. Their evaluator functions are implicit: Secondary characteristics evaluate to true if backward chaining from the secondary characteristic to true-valued primary characteristics succeeds, false otherwise.

The mapping between secondary and primary characteristics is achieved by means of If-Then rules. Each link in the design characteristics hierarchy corresponds to a rule. For example, the following rules link the secondary characteristic is-private to the primary characteristic has-path-with-at-least-2-changes-in-direction-of-more-than-.2-radians:

```
IF [is-hidden x]
THEN [is-private x]

IF [has-circuitous-path-from 'front-door x]
THEN [is-hidden x]

IF [has-path-with-at-least-2-changes-in-direction-of-more-than-.2-radians x y]
THEN [has-circuitous-path-from x y]
```

[5] When computing a path to a territory, the center of the territory is used as an endpoint.
[6] The Joshua software system, by Symbolics Inc., was used.

The user invokes the evaluation mechanism by supplying the name of a design characteristic and applicable design elements, territories, use-spaces, or the name of a design as arguments to a top-level evaluation function. To evaluate the Gale design for the presence of a private main living space, for example, the user types:

```
(has-characteristic private-main-living-space GALE)
```

The result is a textual explanation derived from the backtrace of the rule firings, the value (true or false), and the original backtrace. See Figure 4 for example output taken from the Prairie house experiment described in the next section.

```
The following Prairie secondary design characteristics are present in GALE:

      1. The main living space is private:

         The design has a main living space <LIVING>.
            and
         The space <LIVING> is private:
                 The space <LIVING> is hidden:
                         The front door does not open into <LIVING>.

         The design has a main living space <LIVING>.
            and
          The space <LIVING> is private:
                The space <LIVING> is hidden:
                    There is a circuitous path from the front door to <LIVING>.
                ...
```

Figure 4. Portion of evaluation output for the Mrs. Thomas Gale house; <LIVING> denotes a particular use-space[7] named LIVING. A main living space for a design may be derived or user-supplied.

A final note about territories and use-spaces in the current implementation: Territories may overlap, use-spaces may not. This restriction simplifies the semantics and computation of topological relationships such as adjacency and geometric relationships such as distance or paths between use-spaces. Future versions of the system will remove the restriction.

4. Experiment

As a test of the evaluation system, six secondary characteristics, representing experiential qualities, for Frank Lloyd Wright Prairie houses were encoded, along with fifteen primary characteristics representing geometric details of two-dimensional floorplans. Six Frank Lloyd Wright Prairie houses were evaluated, along with six non-Prairie houses and three Frank Lloyd Wright houses considered to be transitions between his pre-Prairie and Prairie periods. As an exercise, a measure of Prairieness-the number of characteristics present-

[7] The terms use-space and space' will be used interchangeably.

was derived for each design. (See Appendix for the list of design characteristics, designs, and results.)

Frank Lloyd Wright was chosen because he was prolific, has been well-studied, and is regarded as a master at manifesting experiential qualities in his buildings. His Prairie houses were chosen because they share many common features while also being quite varied, and because they have been extensively studied by architectural critics and historians (e.g. Manson, 1958; Brooks, 1972; Twombly, 1979; Hildebrand, 1991), and by researchers interested in computational systems for design (e.g. Koning and Eizenberg, 1981; Chan, 1992). Six representative Prairie houses were chosen, one from each of six Prairie house categories (Pinnell, 1990). The non-Prairie examples were chosen in order to minimize the differences that might be attributed to non-architectural issues. The examples were limited to single-family stand-alone houses, to minimize differences due to type of design; to approximately the same time period, to minimize differences due to societal changes, e.g. addition of a garage; to those about the same size, to minimize differences due to mismatch in number or sizes of spaces; to American designs, to minimize cultural influences. Examples considered transitions between Wright's Prairie and pre-Prairie periods were chosen to see whether the transition nature of the designs would be reflected in the evaluation.

The Prairie houses were evaluated as expected, with all but one exhibiting all six secondary characteristics; half exhibited all fifteen primary characteristics, with the rest exhibiting at least twelve. The transition houses were not distinguishable from either the Prairie examples or the non-Prairie examples; their differences were not captured by the design characteristics used in the experiment. The non-Prairie houses were evaluated as expected with respect to the primary characteristics: Only one exhibited ten of the fifteen; the rest exhibited nine or fewer. The non-Prairie houses and transition houses, however, exhibited many of the secondary characteristics. This result is unsurprising: Focusing on primary characteristics, which represent concrete details, yields a better measure of Prairieness, or perhaps any architectural type, since secondary design characteristics may be manifested through a variety of concrete means, with particular means favored by individual designers. In other words, different architects and their clients may want many of the same experiential qualities in their designs, but may prefer different means for achieving those qualities.

The experiment showed that a system can be built that evaluates an architectural design using knowledge about manifesting experiential qualities in physical form. It also showed that such a system can be used to represent and experiment with definitions of architectural type. It showed that systems can be built that bridge the gap between knowledge-based tools and CAD tools by representing and reasoning with both geometric and nongeometric knowledge.

The experiment also illustrated an obvious limitation in using boolean-valued characteristics for evaluation: How well a design exhibited a particular characteristic cannot be measured. Main living spaces of the non-Prairie examples were deemed private by the system, for example, but in reality are less private than those of the Prairie examples. With boolean-valued characteristics the system cannot make this comparison, nor can it rank designs or make suggestions about how to increase or decrease the values of characteristics. The next system will extend the representation of design characteristics to alleviate these shortcomings.

5. Next

The first version of the system evaluates a design, answering the question: Does this design exhibit a given set of design goals? A design goal is represented as a design characteristic; the desired value of the characteristic is implicitly represented as true. The next version of the system will represent a design goal as a design characteristic plus an explicit value, which may be boolean, qualitative, or quantitative. Quantitative characteristics such as visual openness will be added, providing alternative, and in some cases more precise, methods for evaluating such experiential qualities as privacy. Quantitative characteristics also will facilitate the ranking of design alternatives.

The second version of the system is to include modification, answering the question: How could a design be modified so that it adheres to a set of design goals? To add this capability new functionality must be implemented.

Non-opaque evaluator functions for design characteristics are needed so that derivation of values can be explained in detail. If the system is to suggest modifications to a design in order to change the value of a design characteristic, it must first know why the value failed to be the desired value. For example, if a territory is deemed not private because there is no circuitous path leading to it from the front door, the system must know something more than that the characteristic has-path-with-at-least-2-changes-in-direction-of -more than-.2-radians was false. It has to know why it was false: Were there not enough changes in direction? Were the changes in direction not great enough? For the system to know this, the evaluator function for this characteristic can no longer be an opaque black box that just returns true or false; it must explicitly represent the number and magnitude of changes in direction in such a way that the system can infer corrections. The reasoning will be similar for non-boolean-valued characteristics as well. In reasoning about visual openness, for example, if the value is not as desired, the system will need to know how to change the value. The evaluator in this case must return a quantitative value and give clues about changes in a design that might change the value in the desired direction. The system, therefore, needs an explicit representation of influences on design characteristics, along with knowledge about the directions of influence. Such a representation will be similar to representations used in the qualitative reasoning literature (e.g. Faltings and Struss, 1992).

Reasoning about trade-offs between conflicting goals needs to be added. In trying to satisfy more than a single design goal, the system will have to reason about trade-offs between conflicting goals. The difficulty here is in determining the independence of the various goals. Does increasing the continuity between two territories affect the privacy of one with respect to the other? Does increasing the size of a territory affect the circuitousness of a path from the front door to that territory? Each design goal will need to be translated into a common vocabulary that refers to the design objects to be manipulated so that the effects of simultaneously satisfying particular goals can be judged. The constraint-based reasoning and planning literature contain examples of dealing with these issues (e.g. Allen, et al., 1990; Freuder and Mackworth, 1994).

6. Related Work

Many computer-based systems have been developed for evaluating various aspects of architectural designs. Most of these systems differ from the present work by not evaluating a design with respect to experiential qualities and/or by not operating on geometric representations of physical form. Many of the systems focus on engineering rather than experiential aspects, some employing well-known algorithmic metrics (e.g. Radford and Gero, 1988; Wiezel and Becker, 1992), others employing knowledge-based methods or simulation techniques (e.g. Shaviv and Peleg, 1991; Flemming and Mahdavi, 1993). Some of these systems represent a design's physical form explicitly as physical design elements having locations and dimensions; others instead represent a design only as a set of spaces. Evaluation systems that focus on non-engineering aspects often evaluate spatial organization, but not experiential aspects of the organization. Some of these systems represent a design's physical form explicitly (e.g. Carrara, et al., 1994); others only represent a design as a set of statements derived from a graphical representation of design elements (e.g. McCall, et al., 1990; Oxman, 1992).

A few systems have evaluated designs with respect to experiential qualities, but have not represented or reasoned about physical form, relying instead, for example, on human evaluators and statistical scoring techniques (e.g. Mortola and Giangrande, 1991). One such system (Cao and Protzen, 1994) goes a step further and explicitly represents mappings between experiential qualities and physically measurable properties, but relies on previously collected data rather than measuring properties dynamically from a representation of a design. Several case-based reasoning systems have represented experiential qualities, often derived from postoccupancy evaluations, as annotations, but have not related them to physical form or evaluation per se (e.g. Domeshek and Kolodner, 1992). One case-based reasoning system represents physical form and dynamically calculates values of design characteristics, hinting at the possibility of evaluating experiential characteristics of a design (Dave, et al., 1994). A topological evaluation technique that relates spatial organization to social behavior has been successfully paired with a geometric analysis of visibility (Hanson, 1994). Finally, a recent paper (van der Voordt, et al., 1997) suggests relating physical form to experiential qualities, as is done in work described in this paper.

Several systems are close in spirit to the current work. McLaughlin's system (McLaughlin, 1991) targets the same design task as the current work: evaluation of a developing design with respect to experiential qualities represented as design goals such as open, sunny, and private. The system is a rule-based backward-chaining system that relies on design recommendation literature for knowledge about how to achieve particular design goals in physical form. The system takes as input a design and outputs lists of goals achieved and goals not achieved, along with explanations. The major difference between this system and the current work is that McLaughlin's system does not represent or reason about physical form. The system instead represents a design as a topological arrangement of design elements-walls, windows, doors, openings, spaces-and reasons about the design using a fact base of assertions, as do most rule-based systems. Addition of a geometric representation of a design would simplify some computational tasks (e.g. computation of circulation paths), enable others (e.g. computation of visual barriers), and facilitate interaction with a designer (e.g. by enabling integration of a sketching system).

Two other systems are similar to the current work in their attempts to represent and reason about experiential qualities and physical form. They differ from the current work in ways that result from adopting different views of the design process.

Galle´s system (Galle, 1994) aims to be a general design support tool, facilitating sketching and development of evolving designs. Its goals are similar to those of the design support system envisioned in the opening scenario of this paper. The design knowledge in Galle´s system is that embodied in Alexander's pattern language (Alexander, et al., 1977), which represents general design principles as patterns, arrangements of physical design elements in service of particular design goals. As Galle points out, the patterns "can be of a technical, aesthetic, or social nature." The patterns of a social nature describe methods for achieving experiential qualities. One pattern, for example, suggests achieving what Alexander calls an intimacy gradient by creating a sequence of spaces arranged according to degrees of privacy, with least private near the entrance, followed by slightly more private spaces, leading eventually to the most private. Galle´s system represents patterns and parts, which correspond to physical design elements such as walls, and distinguishes between types and instances. (Pattern types correspond to Alexander's patterns; pattern instances correspond to part instances arranged in the configuration specified by a particular pattern.) Patterns are represented as constraints on user-specified configurations of design parts. Sketching a design amounts to creation of pattern instances: Each part instance added to a design must be associated with at least one pattern instance.[8] If a part is moved, changes propagate to other parts so that all pattern instances associated with the moved part remain satisfied. The current work takes a different view of design: It aims eventually to support sketching, parsing of the sketch into design elements, followed by evaluation of the design with respect to design goals. Design elements will not have to be associated with a particular design goal; they will only be implicitly related to a design goal if they contribute to satisfaction of that goal. Further, the design goals may be stated in terms of secondary design characteristics, which correspond to Galle´s (and Alexander's) patterns, or in terms of primary design characteristics such as lengths of walls or number of turns in a particular path. Primary design characteristics have no user-accessible counterpart in Galle´s current system.

Gullichsen and Chang (Gullichsen and Chang, 1985) built a design generation system based on Alexander's pattern language. It is a rule-based backward-chaining system that relies on the interconnections between patterns, which are ordered in 'decreasing morphological importance to ensure that a whole, imprecisely-specified form is successively differentiated during the process of design." They adhere to Alexander's position that because of this ordering, conflicts do not arise and backtracking during the design process is unnecessary. The system, thus, generates designs in a top-down fashion, progressing from general patterns to more specific patterns which implement general ones. The user initiates the generation by specifying the list of patterns to be satisfied. Gullichsen and Chang's system is concerned with generation rather than evaluation, but parts of it can still be compared with the current work. Their system's view of design as a top-down process differs from the current work's view of design as an iterative process of evaluation and modification. Their systems representation of patterns as predicates in a rule-based system is similar to the current system's representation of design characteristics. Finally, their system's geometric representation of a design may be similar to the current

[8] Galle suggests in a footnote that later versions of his system will relax this restriction.

work's, since mention is made of "lower-level procedures [that] typically employ geometric methods.' An example is given of computing positive space between buildings by calculating the "sum of areas enclosed by walls of buildings or segments which constitute the convex polygonal hull of the building's wings, weighted by the ratio of its enclosing perimeter of the hull," but no mention is made of the underlying representation.

7. Conclusion

The system described in this paper evaluates an architectural design using the designer's theories about how to manifest experiential qualities in physical form. It engages a designer in a conversation of the sort: Does a design adhere to a set of specified design goals? The next system, under development now, will add to this conversation: If not, how can it be modified so that it does? This conversation is an integral part of design; it occurs as a designer searches for a design solution by exploring the relationships between changes in a design and changes in the values of design characteristics. Current work focuses on extending the present system to include more experiential design characteristics, more non-boolean-valued design characteristics, a representation for influences on design characteristics, and a more general representation for design goals. Work also focuses on representations for architectural knowledge about how to "fix" a design, and on reasoning mechanisms that employ the knowledge to explore trade-offs between conflicting design goals. The knowledge is being acquired through discussion with architects.[9] The resulting system will make suggestions for redesign which will produce a design exhibiting as many specified design goals as possible. Such a system will be a step toward an intelligent computer assistant as design conversation partner.

Acknowledgments

The author thanks the following people for helpful conversations, suggestions, and arguments: John Aspinall, Randall Davis, Aaron Fleisher, Mark Gross, Duncan Kincaid, Richard Krauss, Tomás Lozano-Pérez. The author also thanks Bill Mitchell and Patrick Winston for discussion and recommendations that ultimately led to the National Science Foundation Graduate Fellowship funding this work.

References

Alexander, C., Ishikawa, S., Silverstein, M., Jacobsen, M., Fiksdahl-King, I., and Angel, S. (1977) *A Pattern Language,* Oxford University Press, New York.

Allen, J., Hendler, J., and Tate, A., eds. (1990) *Readings in Planning,* Morgan Kaufmann, San Mateo, CA.

Bobrow, D. B., DeMichiel, L. G., Gabriel, R. P., Keene, S. E., Kiczales, G., and Moon, D. A. (1988) "Common lisp object system specification," X3J13 document 88-002R.

Brooks, H. A. (1972) *The Prairie School,* Norton, New York.

Cao, Q. and Protzen, J.-P. (1994) "Deliberation and aggregation in computer-aided performance evaluation," in *Automation Based Creative Design*, A. Tzonis and I. White, eds., Elsevier, New York, 251-264.

[9] Machine learning programs may be of assistance here, though they are outside of the scope of the present inquiry.

Carrara, G., Kalay, Y. E., and Novembri, G. (1994) “Knowledge-based computational support for architectural design,” in *Knowledge-Based Computer-Aided Architectural Design*, G. Carrara and Y. E. Kalay, eds., Elsevier, New York, 147-201.

Chan, C.-S. (1992) “Exploring individual style through Wright's designs,” *Journal of Architectural Planning and Research*, vol. 9, 207-238.

Coyne, R. D., Rosenman, M. A., Radford, A. D., Balachandran, M., and Gero, J. S. (1990) *Knowledge-Based Design Systems,* Addison-Wesley, Reading, MA.

Dave, B., Schmitt, B., Faltings, B., and Smith, I. (1994) “Case-based design in architecture,” in *Artificial Intelligence in Design '94*, Proceedings of the Third International Conference on Artificial Intelligence in Design, J. S. Gero and F. Sudweeks, eds., Kluwer, Norwell, MA, 145-162.

Domeshek, E. A. and Kolodner, J. L. (1992) “A case-based design aid for architecture,” in *Artificial Intelligence in Design '92,* Proceedings of the Second International Conference on Artificial Intelligence in Design, J. S. Gero, ed., Kluwer, Norwell, MA, 497-516.

Faltings, B. and Struss, P., eds. (1992) *Recent Advances in Qualitative Physics*, MIT Press, Cambridge, MA.

Flemming, U. and Mahdavi, A. (1993) “Simultaneous form generation and performance evaluation: a `two-way' inference approach,” in *CAAD Futures '93*, Proceedings of the Fifth International Conference on Computer-Aided Architectural Design Futures, U. Flemming and S. Van Wyk, eds., North-Holland, New York, 161-174.

Freuder, E. C. and Mackworth, A. K., eds. (1994) *Constraint-Based Reasoning,* reprint of Artificial Intelligence 58 (1992) 1-3, MIT Press, Cambridge, MA.

Galle, P. (1994) “Computer support of architectural sketch design: a matter of simplicity?,” *Environment and Planning B*, vol. 21, 353-372.

Gero, J. S. and Sudweeks, F., eds. (1996) *Artificial Intelligence in Design '96,* Proceedings of the Fourth International Conference on Artificial Intelligence in Design, Kluwer, Norwell, MA.

Gross, M. D. (1996) “The Electronic Cocktail Napkin--a computational environment for working with design diagrams,” *Design Studies*, vol. 17, 53-69.

Gullichsen, E. and Chang, E. (1985) “Generative design architecture using an expert system,” *The Visual Computer*, vol. 1, 161-168.

Hanson, J. (1994) “'Deconstructing' architects' houses,” *Environmental and Planning B*, vol. 21, 675-704.

Hertzberger, H. (1993) *Lessons for Students in Architecture*, 2nd ed, Uitgeverij 010 Publishers, Rotterdam.

Hildebrand, G. (1991) *The Wright Space: Pattern and Meaning in Frank Lloyd Wright's Houses,* University of Washington Press, Seattle.

Kincaid, D. S. (1997) (in progress), MArch Thesis, Department of Architecture, MIT.

Koning, H. and Eizenberg, J. (1981) “The language of the prairie: Frank Lloyd Wright's prairie houses,” *Environment and Planning B*, vol. 8, 295-323.

Manson, G. C. (1958) *Frank Lloyd Wright to 1910: The First Golden Age,* Van Nostrand Reinhold, New York.

McCall, R., Fischer, G., and Morch, A. (1990) “Supporting reflection-in-action in the Janus design environment,” in *The Electronic Design Studio*, M. McCullough, W. J. Mitchell, and P. Purcell, eds., MIT Press, Cambridge, MA, 247-259.

McLaughlin, S. (1991) “Reading architectural plans: a computable model,” in *CAAD Futures '91*, Proceedings of the Fourth International Conference on Computer-Aided Architectural Design Futures, G. N. Schmitt, ed., Vieweg, Wiesbaden, 347-364.

Mortola, E. and Giangrande, A. (1991) “An evaluation module for 'An Interface for Designing' (AID): a procedure based on trichotomic segmentation,” in *CAAD Futures '91*, Proceedings of the Fourth International Conference on Computer-Aided Architectural Design Futures, G. N. Schmitt, ed., Vieweg, Wiesbaden, 139-154.

Oxman, R. (1992) “Multiple operative and interactive modes in knowledge-based design systems,” in *Evaluating and Predicting Design Performance*, Y. E. Kalay, ed., Wiley, New York, 125-143.

Pham, D. T., ed. (1991) *Artificial Intelligence in Design*, Springer-Verlag, New York.

Pinnell, P. (1990) “Academic tradition and the individual talent: similarity and difference in the formation of Frank Lloyd Wright,” in *Frank Lloyd Wright: A Primer on Architectural Principles*, R. McCarter, ed., Princeton Architectural Press, New York, 19-58.

Radford, A. D. and Gero, J. S. (1988) *Design By Optimization in Architecture, Building, and Construction,* Van Nostrand Reinhold Company, New York.

Shaviv, E. and Peleg, U. J. (1991) “An integrated KB-CAAD system for the design of solar and low energy buildings,” in *CAAD Futures '91*, Proceedings of the Fourth International Conference on Computer-Aided Architectural Design Futures, G. N. Schmitt, ed., Vieweg, Wiesbaden, 465-484.

Storrer, W. A. (1993) *The Frank Lloyd Wright Companion,* University of Chicago Press, Chicago. Also available as *The Frank Lloyd Wright Companion CD-ROM,* Prairie Multimedia, Inc., West Chicago.

Twombly, R. C. (1979) *Frank Lloyd Wright: His Life and His Architecture,* Wiley, New York.

van der Voordt, T. J. M., Vrielink, D., and van Wegen, H. B. R. (1997) “Comparative floorplan-analysis in programming and architectural design,” *Design Studies*, vol. 18, 67-88.

Wiezel, A. and Becker, R. (1992) “Integration of performance evaluation in computer-aided design,” in *Evaluating and Predicting Design Performance*, Y. E. Kalay, ed., Wiley, New York, 171-181.

Zeisel, J. and Welch, P. (1981) *Housing Designed for Families: A Summary of Research,* Joint Center for Urban Studies of MIT and Harvard University, Cambridge, MA.

Appendix

DESIGN CHARACTERISTICS

The following secondary characteristics, representing experiential qualities, were used in the Prairie house experiment.

1. The design exhibits Wrightian group togetherness.
2. The design exhibits home/hearth symbolism.
3. The main living space is private.
4. The main living space is a place of refuge.
5. The main living space is a place of prospect.
6. A private exterior space is contiguous with the main living space.

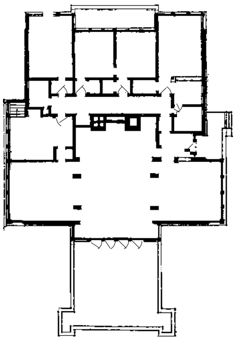
The following primary characteristics, representing geometric properties, were used in the Prairie house experiment. The secondary characteristics to which each primary characteristic contributes are shown in parentheses. (The numbers correspond to those in the above list.)

1. The design has a main living space that is the largest living space. (1)
2. The design has a main living space containing a region from which all other living spaces are visible. (1)
3. The design has a main living space that is connected to all other living spaces. Two spaces are connected if they are no more than one space apart or if they have axially aligned doorways. (1)
4. The design has one fireplace location. (2)
5. The design has a fireplace on an interior wall.
6. The design has a fireplace in the main living space. (2)
7. The path from the front door to the private area does not pass within five feet of the center of the main living space. (3)
8. The front door does not open into the main living space. (3,4)
9. The front door and the main living space are on different levels. (3,4)
10. The path from the front door to the main living space contains at least two changes in direction of greater than .2 radians. (3,4)
11. The path from the exterior approach point to the main living space contains at least two changes in direction of greater than .2 radians. (3,4)
12. The main living space is elevated above the terrain. (5)
13. An exterior space at least 40% of the size of the main living space is contiguous with the main living space. (5)
14. The front door does not open into the exterior space contiguous with the main living space. (6)
15. The path from the exterior approach point to the front door does not pass through the exterior space contiguous with the main living space. (6)

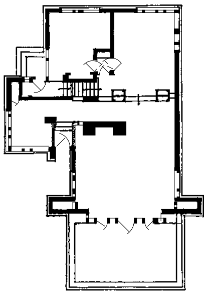
The design examples used in the Prairie house experiment are shown on the next pages. Each floorplan is accompanied by counts of design characteristics exhibited. The first number is the count of secondary characteristics; the second is the count of primary characteristics.

DESIGN EXAMPLES

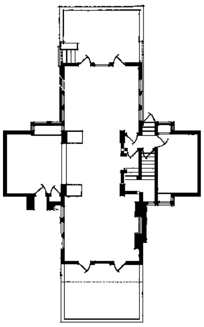
Prairie Houses: Cheney, Gale, Homer, Tomek, Willits, Roberts (Stoner, 1993)



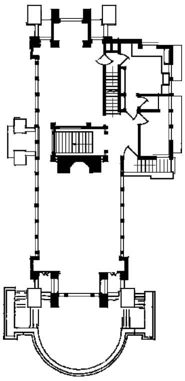
Cheney: 6, 13



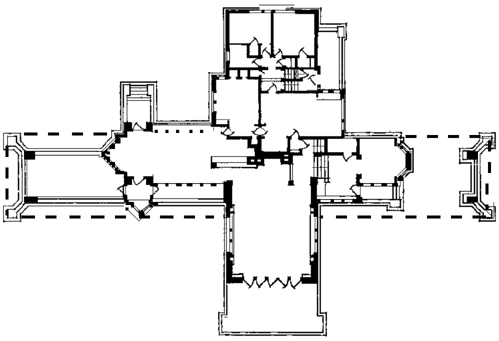
Gale: 6, 15



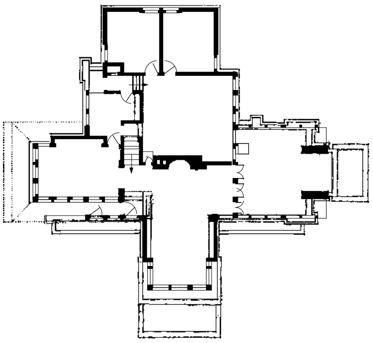
Horner: 6, 14



Tomek: 6, 15

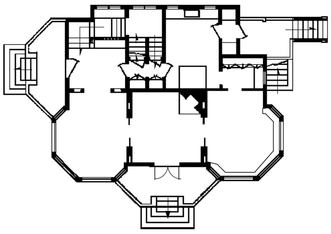


Willits: 5,12

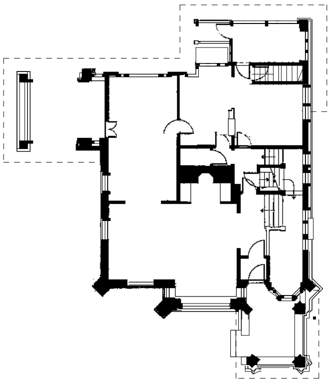


Roberts: 6, 15

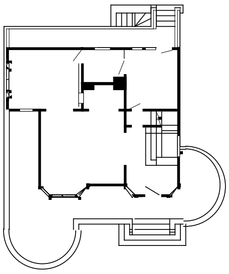
Transition Houses: Emmond, Furbeck, Wright (Storrer, 1993)



Emmond: 5, 13

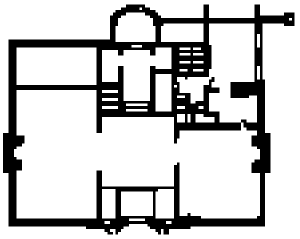


Furbeck: 4, 7

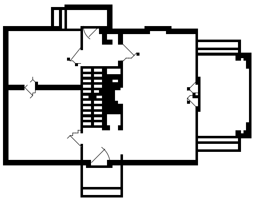


Wright: 4, 7

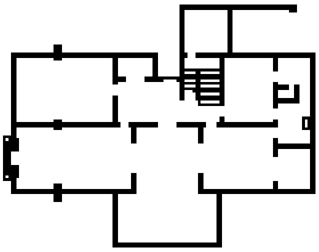
Non-Prairie Houses: Colvin, by George Maher (1916); Jones 5A24 (Jones, 1987); Lawson, by Bernard Maybeck (McCoy, 1975); Mallory, by Arthur Rich (Scully, 1971); Stickley 91 (Stickley, 1982); Winslow, by Frank Lloyd Wright, (Storrer, 1993)



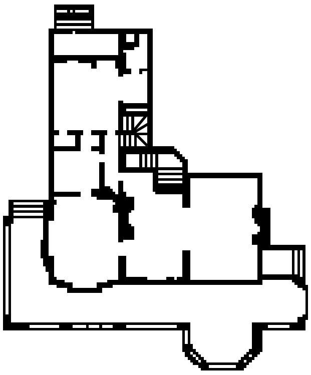
Colvin: 5, 9



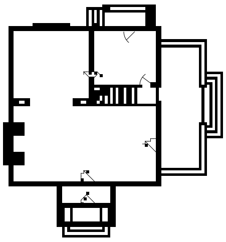
Jones 5A24: 5, 9



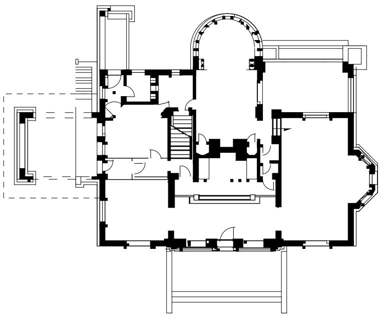
Lawson: 3, 5



Mallory: 5, 9



Stickley 91: 6, 10



Winslow: 4, 6

Sources of design examples:

(1916) *Architectural Record*, vol. 39, 175.

Jones, R. T. (1987) *Authentic Small Homes of the Twenties: Illustrations and Floorplans of 254 Characteristic Homes*, Dover Publications, New York.

McCoy, E. (1975) *Five California Architects*, Praeger, New York.

Scully, V., Jr. (1971) *The Shingle Style and Stick Style; Architectural Theory and Design from Richardson to the Beginnings of Wright*, Yale University Press, New Haven, CT.

Stickley, G. (1982) *More Craftsman Homes: Floor Plans and Illustrations for 78 Mission Style Dwellings*, Dover Publications, New York.

Storrer, W. A. (1993) *The Frank Lloyd Wright Companion*, University of Chicago Press, Chicago. Also available as *The Frank Lloyd Wright Companion* CD-ROM, Prairie Multimedia, Inc., West Chicago.