MULTI-MODAL REPRESENTATION OF DESIGN KNOWLEDGE

Gianfranco Carrara, "La Sapienza" Universily, Rome, Italy
Yehuda E. Kalay State University of New York at Buffalo, USA
Gabriele Novembri, Cartesiana Consortium, Rome. Italy

Explicit representation of design knowledge is needed if scientific methods are to be applied in design research, and if computers are to be used in the aid of design education and practice. The representation of knowledge in general, and design knowledge in particular, have been the subject matter of computer science, design methods, and computer-aided design research for quite some time. Several models of design knowledge representation have been developed over the last 30 years, addressing specific aspects of the problem. This paper describes a different approach to design knowledge representation that recognizes the multi-modal nature of design knowledge. It uses a variety of computational tools to encode different kinds of design knowledge, including the descriptive (objects), the prescriptive (goals) and the operational (methods) kinds. The representation is intended to form a parsimonious, communicable and presentable knowledge-base that can be used as a tool for design research and education as well as for CAAD.

KEY WORDS: Design methods, design process, goals, knowledge representation, semantic networks

KNOWLEDGE REPRESENTATION

In the context of this paper we refer to representation as the *explicit, symbolic expression of some reality or idea for the purpose of communication and presentation.* The symbolic nature of representation distinguishes it from the object or phenomena it designates and frees it from the physical limitations associated with the represented reality itself. At the same time the reality and its representation are linked through some shared human convention or understanding. The representation qualifies, therefore, as a *means of inquiry* into the nature of the represented reality: it allows us to examine and experiment with the subject of the representation as if it were a tangible entity, and through it learn more about the represented reality itself. Additionally, the representation provides a *means of recording* the reality it stands for, in a transferable form.

These characteristics of representation art particularly valuable when the subject of the representation is *knowledge*, which has been the subject of philosophical inquiry since the times of Socrates, Plato and Aristotle. One of the reasons it still remains an enigma is that knowledge is more than the information that is perceived by the senses. Rather, according to Kant [1782], it is *information that was processed by the mind*. The processing extracts the generalizable and useful characteristics of the information and classifies them in a manner that is and applicable in similar future situations. The representation of knowledge, therefore, requires not only the symbolic representation of facts, concepts and beliefs, but also the representation of the processes that *acquire, interpret* and *apply* them in the appropriate circumstances.

The representation of knowledge has been aided considerably by the advent of computers, which art the only device outside the human brain capable of symbolic representation of information along with the processes that operate on it. Specifically, the field of Artificial Intelligence (AI) has been pursuing the issue of knowledge representation (along with its acquisition, interpretation and application) since the mid 1950s and has developed certain tools for this purpose (see for example [Winston, 1977]). More recently, the field of Computer-Aided Design (CAD) has begun to adapt some of these tools for the purpose of representing *design knowledge* (see for example [Kalay, 1987, Coyne et al 1990]).

DESIGN KNOWLEDGE

Design, for the purposes of this presentation, is defined as *a process that specifies the actions that* *ought to be taken in order to achieve certain stated objectives (goals)*. Since there is no formula or procedure that can be used to this effect, design is a process of *exploratory search* where alternative courses of action are hypothesized and their effects me predicted and evaluated by comparing them to the predefined goals. The evaluation may find that the predicted effects achieve the goals, thus validating the proposed course of action. More often, however, the predicted effects of the proposed actions do not achieve the stated goals, or they conflict with each other. In such cases the actions must be modified so they achieve the goals, or the goals must be revised to match the predicted effects of the proposed actions. This is done through iterative adjustment of the proposed actions and the goals until a match is achieved [Simon 1969, Jones 1980, Rowe 1987].

Designers rely on *knowledge* to help them successfully complete this search process. Such knowledge encompasses their own past experiences, aspirations and perceptions, as well as the shared and empirically validated experiences and practices of the discipline as a whole [Tzamir and Churchman 1984]. The subjective, sources of design knowledge include judgement, taste, preferences and feelings, as well as the designer’s subjective view of external realities (his own standard of "truth"). The objective sources of design knowledge include empirical facts gathered through the scientific method in an attempt to discover valid laws and generalizations with predictive power, as well as inferential facts generated through the designer's own conceptual thinking.

An important by-product of the design search process is the enhanced understanding of the problem itself, which is akin to acquisition of new knowledge and the revision of existing knowledge, including the objects, the goals and the methods that link them in the design process [Archea 1987]. Design knowledge, therefore, is a *dynamic entity* that cha-nges as the design process unfolds.

DESIGN KNOWLEDGE REPRESENTATION

The methods that were developed by researchers over the last three decades for the purpose of representing design knowledge generally fall into one of several categories, depending on the researchers' particular needs and the availability of knowledge representation tools. They include constraint-based knowledge representation, issue-based knowledge representation, prototype-based knowledge representation, as well as assorted declarative and procedural knowledge representation tools (shape grammars, expert systems, and procedures) [Galle 1981, Stiny 1981, Carrara & Novembri 1985, Maher & Fenves 1985, McCall 1986, Gross et al 1987, Oxman & Gero 1988, Gero 1990.

The conceptual framework on which we base our approach to design knowledge representation differs from these methods in that it strives to capture multiple *different kinds* of design knowledge, representing both established and evolving design knowledge. Our approach is based on the premise that such knowledge comprises three distinct, yet related, modalities:

1. *Descriptive knowledge*, representing the objects (and concepts) that comprise a particular domain of design, their function and the logical relationships between them (*what* is being designed?).

2. *Nominative knowledge*, representing the goals (intents) a particular design project ought to achieve and the constraints it must abide by (*why* is it being designed?).

3. *Operational knowledge*, representing the methods (strategies) for selecting or generating objects, assigning to them appropriate values and linking them with each other so they meet the specified goals (*how* is it being designed?).

Design knowledge thus comprises a set of attributes, beliefs, perceptions, relationships, aspirations and methods that embody the designer's personal experiences, the profession's shared experiences and the particular circumstances of a specific design project.

Our approach recognizes the differences between the three modalities of design knowledge, and attempts to represent each one of them in the most suitable form rather than fit all of them into a single, albeit powerful, representation. Naturally, such multimodal approach introduces the need to *coordinate* and to *communicate* between the different representational methods, and the need to *classify* design knowledge by the most representative modality (a taskt hat is neither simple nor intuitive).

Two questions must be asked before we present the details of this multi-modal approach. First, are the three modalities *inclusive* of all kinds of design knowledge, or are there kinds of design knowledge that cannot be classified as one (or more) of the three modalities? Second, are these modalities *sufficient* means for design knowledge representation, or are there kinds of design knowledge that they cannot represent? Both questions challenge the scope of the proposed approach and its representational powers. We believe that a complete and provable answer to these questions lies in the domain of general philosophy rather than in the domains of Architecture and CAD, for it is akin to answering the age-old question concerning the nature and scope of human knowledge in the first place. Furthermore, a principal answer can only be found by exercising and applying the proposed approach to many different kinds of design knowledge. Nonetheless, based on our experience of having used the proposed approach for some years, we believe that these modalities are indeed inclusive of both shared and personal design knowledge, and that they can represent the key aspects of the knowledge used by designers.

The following sections describe how we have begun to approach the representation of each kind of design knowledge, and how the different representations can be combined into an overall computational design framework.

DESCRIPTIVE DESIGN KNOWLEDGE

Physical design involves operations on actual and conceptual objects. To represent design knowledge, therefore, it is necessary to represent the objects that are the substance of the designed artifact or environment. Many computational methods have been developed for the purpose of representing objects and concepts. One of the most successful methods is a graph-theoretic formalism known as *semantic networks* [Shapiro 1971, Shapiro 1979]. Like other networks, semantic networks consist of nodes and labeled, directed arcs (Figure 1). The nodes represent individual objects, concepts, and their properties in some specific domain of discourse. Arcs represent binary relationships between the nodes, such as class membership (denoted by the label IS_A), sub-class relationship (denoted by the label A-KIND-OF, or AKO for short), part-whole (assembly) relationship (denoted by the label PART_OF) and property relationship (denoted by the label HAS).

A semantic network represents knowledge in a distributed manner. For example, the knowledge that LAMPS are an artificial source of lighting used in office buildings can only be inferred by examining seven nodes and six arcs in the network depicted in Figure 1. This may seem wasteful compared to simply stating the facts as done in this paragraph. However, the same network also tells us that LAMPS are ACCESSORIES and they are part of ARTIFICIAL LIGHT, which is only one kind of a LIGHTING SOURCE used in the LIGHTING of OFFICE BUILDINGS (the other being DAYLIGHT, which is composed of different kinds of ILLUMINATION, etc.). Hence, the network as a whole conveys much more information about office lighting than this paragraph does, in a parsimonious way that is comprehensible to both humans and machines.

It is the associative nature of the network that makes it an efficient and effective means for representing large quantities of related facts. However, the distributed nature of the representation means that a certain effort must be exercised to retrieve the stored information: the network must be traversed, and the information that is pertinent to understanding some concept must be *inferred* from multiple nodes and arcs.

Frames

Much of the representational power of the semantic network formalism is derived from the flexibility of traversing the network in different ways, and the ability to create new nodes and link them to existing ones. Nevertheless, some relationships are more permanent than others and could benefit from a representational formalism that recognizes their permanency. Such relationships are particularly in evidence between *objects* and their *properties.* Unlike the loose relationship between nodes that represent distinct objects, which may change under the circumstances of different contexts, the relationship between an object and its own properties, such as its cost, the materials it is made of and its dimensions, is fixed (this is not to say that the *values* of the properties themselves are fixed). Such property relationships can be accommodated in a semantic network by expanding the nodes into *frames*.

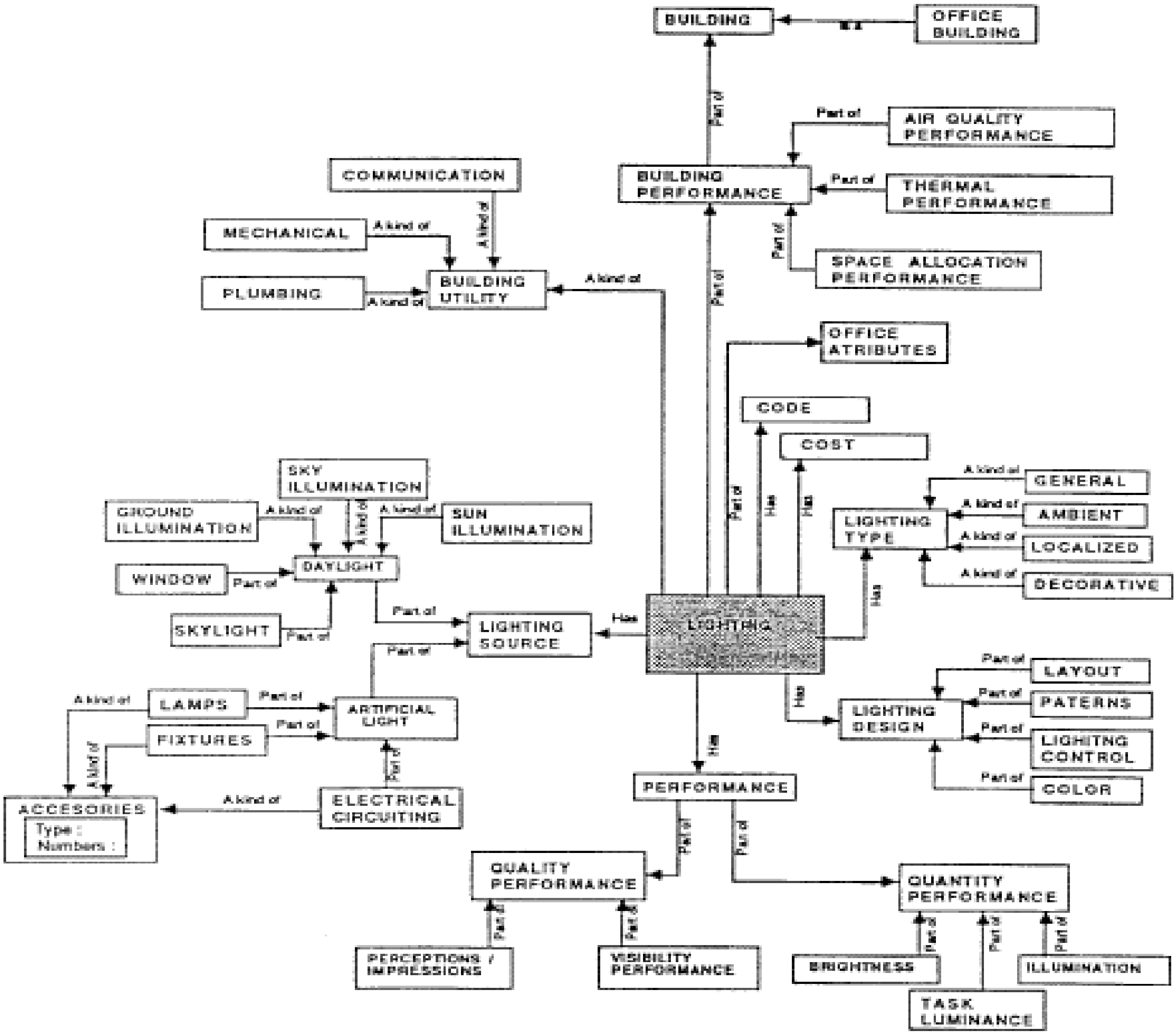

Figure 1: A semantic network that represents descriptive-knowledge about lighting in office buildings (developed by A. Kodijat, 1991)

Frames, which were first introduced by Minsky in the 1970s, transform nodes into DBMS-like records and allow them to include their own set of attributes in the form of name-value pairs. Minsky called the name-value pairs *slots* and *fillers,* respectively. Slots can be thought of as "place holders," or as predefined attributes that can be associated with particular types of values. Fillers are those values. They include the permissible range of values that can be associated with a particular slot, default values and even instructions (so called "demons") that allow the slot to calculate its Filler value when it depends on values associated with other slots. For physical objects of the kind used in design there would typically be a slot called *shape,* whose filler would be a particular geometric entity describing the form of the object and it's location In space relative to some frame of reference. There would also be slots for material composition, structural properties, cost, thermal properties, and so on. Additionally, there would be slots describing the *function* of the object, and how this function is affected by the context in which the object is embedded.

Their standardized structure qualifies frames to be used as *templates* of entities of the kind they describe. Moreover, missing information can be filled-in automatically, in the form of default values, by associating a predefined value with any static (non-procedural) slot. Figure 2 depicts a frame representation of the object DAYLIGHT. It includes such slots as FUNCTION, SKY CONDITION, LOCATION, and so on.

Semantic networks appear to be a suitable means for representing descriptive design knowledge. Implemented by means of an object-oriented programming

DAYLIGHT	
function:	(main, backup, both)
sky condition:	(overcast, clear)
location:	(solar altitude)
orientation:	(north, south, east, west)
site conditions:	(surrounding buildings, landscaping)
accessories:	(window, skylight, atrium, sunshade, blinds)
system:	(direct, indirect)

Figure 2: A frame representation of object DAYLIGHT.

language like C++, they enable us to represent existing design knowledge in a parsimonious and tractable manner, and extend and modify this knowledge by adding new nodes and arcs. New paths can be explored without changing either the nodes or the arcs, reflecting new semantic meanings.

NORMATIVE DESIGN KNOWLEDGE

Design, as stated earlier, can be described as a goal-oriented search process where candidate solutions are produced and examined while seeking one that meets certain desired conditions [Simon 1969, Simon 1972 Akin 1978]. The goal (or more generally, the goals) that the design process attempts to accomplish can, therefore, be regarded as a *normative prescription of the properties* that the product of the design process should posses. Every design process begins with a set of design specifications (since it is intended to accomplish some specific human need or want). These specifications represent the designer's (or the client's) beliefs about what the product of the design ought to be like. They do not describe the product itself, nor the methods that must be used to accomplish them. Furthermore, as the design process unfolds and as the designer gains more information and understanding of the particular opportunities presented by the context and the limitations it imposes, he will modify the specifications to reflect this newly discovered knowledge.

Design goals thus serve to represent the designer's intents in an explicit manner without imposing a particular solution. For example, in designing the lighting in an office building the client's intents of creating a functional environment may . include the creation of certain atmosphere (e.g., privacy), which can be achieved physically by different means (e.g., task lighting combined with ambient lighting). The representation of the *intent,* rather than the specific physical solution, allows the designer to experiment with different solutions that capture not only the physical characteristic-s of the desired lighting but also the psychological effects that the physical solution ought to accomplish. The representation of intents thus serves to explain the intrinsic meaning of a concept, which may not be the same as its extensional manifestation [Kim 1987].

The representation of design intents forms an important part of representing design knowledge. It provides the designer with the necessary flexibility to choose between alternative solutions, negotiate impasses and take advantage of unfolding opportunities. A method for representing design intents must, therefore, be able to capture the full complexity of human wants and needs, intentions and conventions. For the purposes of CAD this representation must be meaningful to both humans and machines.

Given that intents establish a context for tradeoffs and decision-making in the design process, they could be stated as acceptable alternative performance measures, or as the *constraints* that the proposed solution must satisfy. Such constraints fall into two categories:

- *Extentional constraints,* such as gravity, wind resistance, building codes, and so on.
- *ntentional constraints,* such as budget, number of rooms, function, style, and so on.

The first kind of constraints, called "hard" constraints, are either satisfied by the solution or not (they are pass/fall constraints) [Maher & Fenves 1985]. The second kind of constraints, called "soft" constraints, may be satisfied to different degrees. The degree of satisfying a particular constraint of this kind may depend on the degree of satisfying other constraints, thereby introducing the import ant concept of *trade-offs*. For example, the intent of creating an atmosphere of privacy in the office could be accomplished by providing fluorescent or high intensity discharge lamps, depending on the intensity, distribution and layout of the fixtures. Such tradeoffs are the hallmark of architectural design and they depend on judicious decision making on the part of the architect and the client.

Design intents may change as the design process evolves due to discovering new opportunities or impasses offered by the emerging solution, incompatibilities among the design intents themselves and changing priorities of the architect or the client during the design process [Jockusch 1992]. The representation must, therefore, allow for and support the modification, addition and deletion of goals during the design process and consequently, the assessment of the implications of such changes on prior decisions.

Constraints, goals, and goal hierarchies

We have chosen to represent normative design knowledge as *goals,* stated in terms of the *constraints* that must be satisfied by candidate design solutions. Each constraint indicates the level or kind of performance a design solution must achieve in a particular category. The representation is based on the following general notational form:

```
constraint ( value | range ).
```

For example:

```
illumination level (150 fc)
lighting system (ambient)
spatial impression (privacy).
```

A *set* of constraints indicates a particular combination of desired performances that must be achieved by a candidate design solution in order to achleve a specific design intent. We call this set a *goal*. It is based on the following general notational form:

```
goal ( {goal} | {constraint} )
```

For example:

```
Office Lighting
 (Performance ((illumination level (150 fc)),
       (lighting system (ambient)),
       (spatial impression (privacy))),
 (Budget ($2500)),
 (Design (pattern (distributed)),
       (density (medium)),
       (source (high-intensity-discharge)))).
```

There is no inherent difference between goals and constraints. Rather, they form a hierarchical structure where terminal nodes represent constraints and intermediate nodes represent goals, as depicted in Figure 3. The goals are considered satisfied if all their constraints have been satisfied, according to the definitions stored in the semantic network's frames.

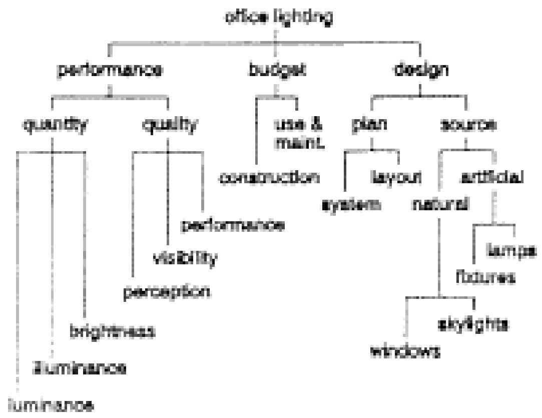

Figure 3: Goal hierarchy of office lighting design.

The representation of tradeoffs

The specificity of design goals must not be confused with the specificity of the design solutions that satisfy them. Different design solutions may achieve the same goal, albeit each may satisfy the constraints comprising the goal differently. The different performance levels at which alternative sets of constraints may be satisfied represent trade-offs in the context of achieving a particular goal. For example, the specifications in the design of office lighting may include performance, budget and design. A design solution that accomplishes this goal would have to meet all these specifications. It is likely, however, that alternative solutions would meet the constraints at different but equally acceptable levels of satisfaction. To accommodate such tradeoffs alternative goals can be formed, each representing the accomplishment of the same intent but reflect different preferences. For example, compare the following goals to the ones stated earlier:

Office Lighting
(Performance ((illumination level (150 fc)),
(lighting system (ambient)),
(spatial impression (privacy))),
(Budget ($1500),
(Design ((pattern (distributed)),
(density (high)),
(source (fluorescent)))).

Together, these two sets of goals represent alternative, acceptable solutions for achieving the intent of designing office lighting. Design goals of the same type thus represent an *equivalence class* of context dependent, related set of constraints which define the conditions that ought to be met by a design solution. The constraints (and hence the goals) do not define the solution itself, only one alternative measure for its acceptability.

The conditions under which various constraints are considered satisfied and the minimal acceptable levels of satisfying each constraint are stated in the semantic network. The particular combination of constraints that is considered a "goal" is established when the goals are first introduced. They may be changed by the designer as the process unfolds and as the circumstances of the particular design are manifested by his ability to accomplish the goals. Additional goals may be added, or existing goals may be deleted from the hierarchy, thereby providing additional measures of flexibility and means to represent changing preferences as the design process evolves.

OPERATIONAL DESIGN KNOWLEDGE

The representation of design objects and goals alone, even through the semantically-rich formalism of semantic networks and goal hierarchies, does not capture all the experiences architects use when searching for design solutions. An important part of design knowledge comes in the form of *methods, procedures,* and *tactics*. Such procedures, which sometimes take the form of manuals, recipes, codes and even design "styles," form an important body of design knowledge that we refer to as *operational design knowledge.*

Operational design knowledge comprises the methods designers use to generate and modify candidate design solutions, and the methods they use to predict and evaluate their expected performances. Two kinds of design methods can be identified and, it is argued here, must be represented:

1. General design methods (meta-methods, or design strategies) that help organize the flow of design decisions.

2. Specific design methods (tactics) for generating and evaluating design solutions.

To the first category belong methods for selecting design goals to be accomplished. To the second category belong methods for accomplishing the selected goals. The addition and modification of goals also forms part of general design methods, and means for evaluating candidate design solutions form part of the specific methods.

General methods

Much has been written about meta-design methods, which are often referred to as *planning* [Gero & Coyne 1987, Mitchell 1990]. Typically, a path leading from general goals to specific goals (called *top-down*) or from specific goals to general ones (called *bottom-up)* is assumed. It has also been argued that no specific direction should be assumed, for designers follow the design search path that is most suitable for their problem [Kodijat 1991]. Our aim here is not to critique any of these methods, but rather to argue that they should be represented explicitly so they can be studied and improved upon.

Regardless of the strategy being followed, designers may choose to achieve a particular goal by following some pre-planed course of action, or by selecting one at will. Alternatively, they may choose to add new goals. We refer to this process as *goal selection* (Figure 4).

Specific methods

Goal-achievement refers to choosing the actual process for developing or evaluating a design solution to meet the constraints listed in the selected goal (Figure 5). This process also includes the ability to

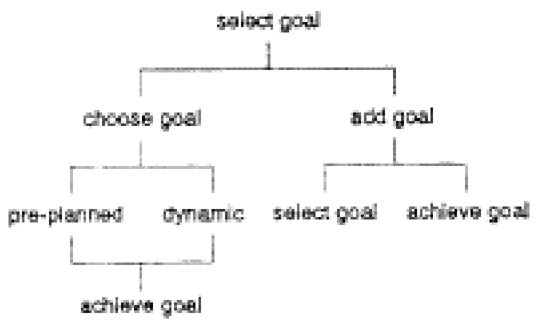

Figure 4: Schema of design goal selection.

modify goals, as discussed earlier. Design methods of this kind can be expressed as rules, algorithms and other means derived from the experience of individual designers, the cumulative experience of the profession as a whole, or from building codes and regulations (such as fire egress codes, design for accessibility, etc.). Many specific methods have been developed for a wide range of design tasks, including generators of schematic floor plans, automatic generation of bills of materials, generation of staircases, generation of kitchen layouts, the evaluation of area, volume, energy, cost, wayfinding, habitability and many more building performances. Each method typically comprises a well-defined body of knowledge represented in some computational form. They have been represented by such methods as prototype refinement, shape grammars, expert systems and procedures [Mitchell 1977, Kalay 1987, Coyne et al 1990, Kalay 1992, McCullough et al 1990].

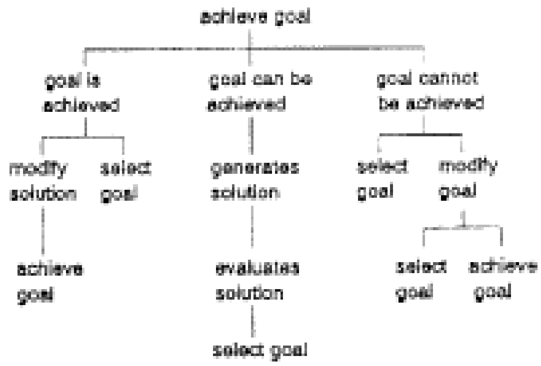

Figure 5: Schema of design goal achievement.

Our approach thus advocates an open-ended integration that permits the addition, replacement or modification of particular solution generators and evaluators as they become available. The use of design goals permits, among other things, the communication of the results (or at least the implications) among different design generators, and the evaluation of candidate design solutions from many points of view. Thus, for instance, if a design method exists that can generate an architectural floorplan using room sizes and adjacency relationships as its guiding rules (e.g., [Schwarz 1992]), then its results could be evaluated by methods that are concerned with habitability, energy, cost, constructability, and other performance measures [Fenves et all 1992]. The results are compared to the constraints specified in the goals, and perhaps modified to better meet all the requirements, even if they become less optimal from the particular generator's point of view.

HOW IT ALL COMES TOGETHER

Although this paper is intended only to discuss the representational issues concerning design knowledge, it will be incomplete without outlining how the different representation are used in support of each other in the context of the design process. To do so we must connect the different representational modules, and augment the representation with operators that manage and coordinate the flow of information between them. We propose to combine the knowledge representation modules in a framework of the kind depicted in Figure 6.

This framework includes, in addition to the three components discussed earher, also a *user interface and design process control* component, which provides the "glue" that binds the different knowledge representations into one functional whole. The nature of this interface depends on the particular tools used to represent the knowledge, and the environment in which the system is implemented. It will surface to say that this component must support both graphical and logical communication with the user, and it must have certain intelligence of its own to know which module to invoke at what time.

Perhaps the best way to illustrate how we envision this framework to function in the design process is by way of an hypothetical example, drawn from the domain of designing office lighting (Figure 7).

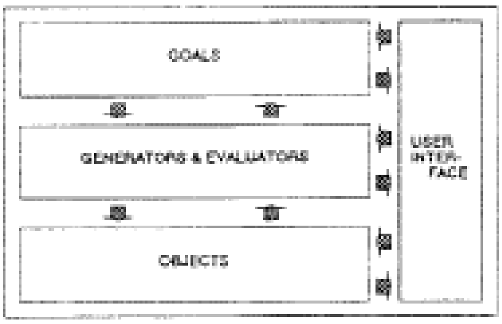

Figure 6: A framework for integration.

CONCLUSIONS

The central thesis presented in this paper is that design knowledge can and should be represented explicitly, and that it comprises different modalities that should be represented differently (objects, goals, and methods), The reasons for undertaking the considerable effort of representing design knowledge in this form are threefold:

1. By capturing in an objective and explicit manner the knowledge architects (as well as other designers) use to accomplish their tasks, we can study these methods and possibly improve upon them.

2. The representation itself will form a transferable, teachable body of knowledge, thus contributing to the education of new generations of architects.

3. The knowledge, which is represented in a form comprehensible to computers, forms the basis for developing intelligent design tools that could help architects better accomplish their tasks.

A particular set of computational knowledge representation tools was presented, which we believe can meet the challenges presented by the stated objectives. Many of these methods have been developed by researchers in other disciplines, for different purposes than our own, and had to be adapted to the needs stated here. It is our belief that their particular combination, as presented here, is mutually supportive and complementary, and together they can better respond to the challenge.

Much more work remains to be done to validate or disprove the particular knowledge representation hypothesis that was presented here and to develop the set of tools that can implement it. Even more work will be needed to capture the knowledge itself, which will transform the proposed approach into a comprehensive design tool capable of supporting designers in their work. Nevertheless, we believe that recognizing the multi-modal nature of design knowledge and outlining the computational means that could be used for its representation are the necessary first step in that direction.

Input	Inferences
office attributes:	
size = 300sqft	
height = 10 ft	
function = executive suite	(from INPUT)
	room type = private office
	(from PRIVATE OFFICE)
	spatial impression = privacy
	or
	spatial impression = pleasant
	or
	spatial impression = relax
	illumination level = 150 fc
	lighting sys. = task lighting
	and
	lighting sys. = ambient lighting
	(from TASK LIGHTING)
	lamps = fluorescent luminaire
	or
	lamps = high intensity discharge
	(from AMBIENT LIGHTING)
	pattern = distributed
construction budget < = 2500	
	(from BUDGET and SIZE and LIGHTING SYSTEM)
	layout = medium distribution

Figure 7: A sample dialog between the designer and the integrated system.

ACKNOWLEDGEMENTS

The ideas and the methods described here owe much to the following individuals: Gianfranco Carrara, Anton Harfmann, Ardito Kodijat, Bruce Majkowski, Gabriele Novembri, Rivka Oxman, Edna Shaviv, Rajeev Subramian and Lucien Swerdloff. These ideas were developed under the auspices of several grants provided through the generosity of several private corporations and the governments of the United States and the State of Israel.

REFERENCES

Akin O. "How do architects design?" *Artificial Intelligence and Pattern Recognition in Computer-Aided Design* (Latombe, ed.), IFIP, North Holland Publishing Co., New York, 1978.

Archea J. "Puzzle Making: What Architects Do When No One is Looking," *Computability of Design* (Kale, ed.), John Wiley & Sons, New York, 1987.

Carrara G. and G. Novembri "Constraint-Bounded Design Search," *Computer-Aided Architectural Design Futures* (Alan Pipes, ed.), pp. 146-157, Butterworth, London 1985.

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

Fenves F.J., U. Flemming, C. Henderson, and M.L. Maher 'Performance Evaluation in an Integrated Software Environment for Building Design and Construction," in *Evaluating and Predicting Design Performance* (Y.E. Kalay, ed.), John Wiley & Sons, New York, 1992.

Galle P. "An Algorithm for Exhaustive Generation of Building Floor Plans," *Communication of the ACM,* Vol. 24 No. 12, pp. 813-825, December 1981.

Gero, J.S. "Design Prototypes: A Knowledge Representation Schema for Design.- *A I Magazine, Vol. 11,* pp. 26-36, 1990.

Gero J.S. and R.D. Coyne "Knowledge-Based Planning as a Design Paradigm," *Design Theory in Computer-Aided Design* (H. Yoshikawa &, A. Warman, eds.), North Holland, Amsterdam, 1987.

Gross M., S. Ervin, J. Anderson and A. Fleisher "Designing with Constraints," *Computability of Design* (Kalay, ed.), John Wiley & Sons, New York, 1987.

Jockusch P.R.A. "How Can We Achieve A Good Building?" in *Evaluating and Predicting Design Performance* (Y.E. Kalay, ed.), John Wiley & Sons, New York, 1992.

Jones C.J. *Design Methods.- Seeds of Human Futures,* John Wiley & Sons, New York 1980.

Kalay Y.E. "Redefining the Role of Computers in Architecture: From Drafting/Modeling to Knowledge-Based Assistants," *Computer-Aided Design Vol.* 1-,7, No. 7, pp. 319-328, September 1985.

Kalay Y.E. (cd.) *Computability of Design,* John Wiley & Sons, New York, 1987.

Kalay Y.E. (ed.) *Evaluating and Predicting Design Performance,* John Wiley & Sons, New York, 1992.

Kant I. *Critique of Pure Reason,* (N. Kemp Smith, Trans.) Random House, New York, 1958 (original work published in 1782).

Kim M.K. "Development of Machine Intelligence for lnference of Design Intent Implicit in Design Specifications," *Computability of Design* (Kalay, ed.), John Wiley & Sons, New York, 1987.

Kodijat A.M. *A Computational Model for the Design of Office Lighting,* Masters Thesis, School of Architecture and Planning. SUNY-Buffalo, 1991.

Maher ML and S.J. Fences "HI-RISE: A Knowledge-Based Expert System for the Preliminary Structural Design of High Rise Buildings," Technical Report R-85-146, Department of Civil Engineering, Carnegie Mellon University, Pittsburgh PA, January 1985.

McCall R. "Issue-Serve Systems: A Descriptive Theory of Design," *Design Methods and Theories,* 20:3, 1986.

McCall R., G. Fischer and A. Morch, "Supporting Reflection-in-Action in the Janus Design Environment," *The Electronic Design Studio* (McCullough et al, eds.), MIT Press, Cambridge MA, 1990.

McCullough M., WJ. Mitchell and P. PurceH *The Electronic Design Studio,* MIT Press, Cambridge MA, 1990.

Mitchell WJ. *Computer-Aided Architectural Design,* Van Nostrand Reinhold, New York, 1977.

Mitchell W.J. *The Logic of Architecture.- Design, Computation, and Cognition, MIT* Press, Cambridge MA, 1990.

Oxman R.E. and J.S. Gero "Designing by Prototype Refinement in Architecture, "*Artificial Intelligence in Engineering Design* (J.S. Gero, ed.), pp. 395-415, Springer Verlag/Elsevier Publications, New York, 1988.

Rowe P.G. *Design Thinking, MIT* Press, Cambridge MA, 1987.

Shapiro S.C. "A Net Structure for Semantic Information Storage, Deduction, and Retrieval," *Proceedings of the Second International Joint Conference of Artificial Intelligence, Vol. 2, pp.* 512-523, Morgan Kaufmann Publishing, Los Altos CA, 1971.

Shapiro S.C. "The Sneps Semantic Network Processing System," *Associative Networks: The Representation and Use of Knowledge by Computers* (N.V. Findler, ed.), pp. 179-203, Academic Press, New York, 1979.

Simon H.A. *The Sciences of the Artificial,* MIT Press, Cambridge MA, 1969.

Sirnon H.A. *Human Problem Solving,* Prentice-Hall, Englewood Cliffs NJ, 1972.

Stiny G. "Introduction to Shape and Shape Grammars," *Environment and Planning B, Vol.* 8, pp. 343-351, 1981.

Schwarz A. *Representing and Solving the Automated Building Design (ABD) Problem,* PhD Dissertation, Department of Computer Science, The Technion IIT, Haifa Israel, 1992.

Tzamir Y. and A. Churchman "Knowledge, Ethics, and Environment--Behavior Studies in Architectural Education," *Environment and Behavior,* 1984.

Winston P.H. *Artificial Intelligence,* Addison-Wesley, Reading MA, 1977.