Design and Semiotics: Proceedings of a Symposium organized by Dr. Mihai Nadin Eminent Scholar Art and Design Technology

UTD AUTHOR(S): Mihai Nadin

Creative Commons 4.0 Attribution License

The Cognitive Condition of Design March 26-27 1987

PROCEEDINGS

Editor: Brian S. Zaff

A Symposium organized by Dr. Mihai Nadin Eminent Scholar Art and Design Technology
On March 26 and 27, 1987 graphic, product and interior designers, architects, city and regional planners, psychologists, computer scientists, students and others interested in the issues surrounding design met to discuss the cognitive condition of design. There was at this symposium no formal presentations of scholarly papers, but rather the free and open exchange of ideas concerning the range of activities that have as their common denominator design.

A wide array of topics were touched upon ranging from the nature of complexity to the proper curriculum for a design education, but the issue that was the primary focus of the inquiry, the issue that served as a common thread weaving its way through the two days of discourse concerned the very nature of design. General questions expressed repeatedly, such as, "what is the nature of design?" and "what is it that designers do when they design?" were contemplated with the expressed intention of evaluating the likelihood that the various aspects of this activity called design could be expressed in formal, computational terms.

In what follows, the reader will be exposed to numerous definitions of the activity of design and numerous predictions concerning the prospects of formalizing all or part of the design process. As one might well expect with such a varied group of participants no general consensus was reached, in fact an impassioned debate persisted for hours after the symposium was adjourned. Thus, it is left to the reader to draw his or her own conclusions about the various topics discussed during this symposium.

The proceedings were prepared from a taped transcript of the symposium, and every effort was made to capture the ideas in their original expression, while transforming the spoken dialogue into a form more easily read. In editing the transcript very little external structure was imposed on the discourse, the intention
being to faithfully preserve the exchange of ideas as they occurred. A number of participants relied heavily on the graphic presentation of ideas, and in spite of every effort, regrettably, with only a few exceptions, the images that were used could not be obtained. Admittedly the text suffers for this, and in several instances substantial sections of a presentation were deleted when they were no longer meaningful in the absence of a graphic referent. What remains is nonetheless an animate discussion of the cognitive condition of design.

Columbus, Ohio
April, 1988

Brian S. Zaff, Editor
## CONTENTS

<table>
<thead>
<tr>
<th>Part</th>
<th>Title</th>
<th>Author(s)</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part I</td>
<td>The Common Denominator: Is There Such a Thing?</td>
<td>Mihai Nadin</td>
<td>1</td>
</tr>
<tr>
<td>Part II</td>
<td>Changing Problems + Changing Capabilities = Changing Design</td>
<td>Patrick Whitney</td>
<td>5</td>
</tr>
<tr>
<td>Part III</td>
<td>Design Language</td>
<td>John Rheinfrank</td>
<td>28</td>
</tr>
<tr>
<td>Part IV</td>
<td>The Cognitive Style of Design</td>
<td>Ronald MacNeil</td>
<td>46</td>
</tr>
<tr>
<td>Part VI</td>
<td>Design as Knowledge-Based Problem Solving Activity</td>
<td>B. Chandrasekaran</td>
<td>117</td>
</tr>
<tr>
<td>Appendix</td>
<td>Directory of Participants</td>
<td></td>
<td>148</td>
</tr>
</tbody>
</table>
PART I

The Common Denominator: Is There Such a Thing?

Introduction by Mihai Nadin
Engineering design, graphic design, genetic design, program or algorithm design, architectural design--I can go on and on--are fields of human activity which seem to share something, since their denotation has "design" as a common denominator. Actually, it looks as though there is no human activity which does not involve a design component. Sure, there is a difference between the design of a business and the design of a computer graphics algorithm, we do not need a symposium to agree on this. What we do not know is how people design and how, in spite of the differences between various forms of design, we are able to distinguish between good and bad design.

In recent months--I can even say in recent days--experts in fields such as artificial intelligence, genetics, linguistics, and mathematics have advanced new hypotheses not in their field of expertise, but in design, acknowledging the importance of design in every form of human creativity. In my preliminary meetings with some of the participants, I was presented with a very challenging definition of design: Design is communication of creativity. I do not take credit for this definition; I expect the person who suggested it to make his own point during the symposium. But I would like to say that the relation between creativity and design seems critical at the level our civilization has reached. It is no longer possible to waste material and energy on poorly designed products. It is no longer possible to substitute quantity for quality, to be competitive while ignoring the contribution of design. It is no longer possible to use scientific developments and new technology without a design concept that integrates science and technology.

One of the contributions made by experts to a better understanding of design comes from Terry Winograd and Fernando Flores (incidentally, Winograd is the author of one of the famous artificial intelligence programs--SHRDLU). The two question our main design concept--representation--and argue that in order to be able to build better computer systems, we need an improved design concept. They
quote from the work of Humberto Maturana, a biologist concerned with understanding how biological processes participate in cognition and language: "Living systems are cognitive systems, and living, as a process, is a process of cognition." (cf. p. 46 in Winograd & Flores, Understanding Computers and Cognition. A New Foundation for Design, 1986.) Indeed, we have to understand that design is not independent of the designer and that design intelligence cannot be reduced to strategies of representation. In another contribution, George Lakoff deals with the role our experience plays. He rejects the idea that reason is some kind of manipulation of abstract symbols and that the realization of meaning is a result of representation. In Lakoff's view, the mind-as-computer metaphor eliminates the role of experience and imagination, the role of metaphors, metonymy, and mental imagery. His book (Women, Fire and Dangerous Things: What Categories Reveal About the Mind, due to appear in May, 1987) takes a look at the mind through the window of the study of language. Let me suggest to you that our symposium is an attempt to look at the mind through the window of design. And I think, due to the nature of design, this window is more intriguing.

I mentioned the two books and alluded to recent articles because in my mind, if we intend to improve the quality of design, we have to better understand what design is. More simply put, we need a theory of design. Enormous amounts of money and much human energy are put into products doomed to fail because of our love affair with technology, we are seduced by the transient, attracted to effects, and transformed into operators. Human creativity deserves better. It is my hope that under the stimulating leadership of Patrick Whitney, Ronald MacNeil, and William Mitchell, you will become active participants in an exchange of ideas, questions, and experiences that might tell us if design is the common denominator of such divergent activities as engineering, programming, architecture, and product development. The rules of this symposium are very simple: No rules apply. Use
every means of expression you are aware of to express yourself. Just don’t abuse your colleagues by indulging in the type of lecture we did not want to have in this program in the first place. Talk and listen. Listen and speak. Use the meeting time and the time between meetings to learn from others and so share with others. I would like to thank all of you for being here. And please bear in mind: You will get out of this symposium as much as you are willing and prepared to get.
PART II

Changing Problems + Changing Capabilities = Changing Design

Discussion led by Patrick Whitney
Patrick Whitney: The focus of this morning's discussion will be relatively non-technical, non-hardware oriented aspects of design. What I intend to address are the pragmatic concerns of the design process, namely, how we as designers work, and how we make things that will be used by other people. I am not referring to the design process in the most general sense, as a creative, problem solving process, but more specifically in regards to how we do our work to make better products, better buildings, better messages, and better interiors for people to eventually use.

Both in the offices of Jay Doblin and Associates in Chicago, a design planning firm, that writes plans for relatively large design projects and at the Institute of Design where I teach, we are very interested in the idea of complexity and how complexity seems to be confusing us as well as a lot of our students, clients and colleagues. By complexity, I mean the things that we see around us and read about in the newspaper everyday. By way of example, I have collected a short list, taken from last week's newspapers. The list consists, not of headlines, but of the topics under which several articles could be grouped; topics which are by now cliches. We are faced with new production processes, smart production processes, automated assembly, and robotics. We are faced with a transformation within the product itself, from dumb products to smart products and with what it means to have microprocesses within the product. We are looking at the constant decay, decentralization and decomposition of the standard model of demographics. We are confronted with the general decline of mass marketing and mass society and witness to the growth of regional products. And, as if to add to the total confusion we see at the same time, an emergence of a large number of world products. Products which can be produced in one country and sold around the world regardless of the specific local culture. We see, for example, major consumer products such as Pampers, Coke, and Marlboro, that are not culturally specific. This pattern of growth in world products is not difficult to understand, for it shadows
the growth of world economies. It is also linked to this idea that many things are becoming centralized, offering as an example, the English language which is becoming in a sense, a standardized means of communication in the advanced world. These tendencies toward centralization add to the complexity, because at the same time, we find massive tendencies towards decentralization in other areas of human activity. So, we see an increase in world products and increase in regional products. An increase in centralization and an increase in decentralization. It is very hard to discern the patterns of these changes.

It seems to me, that it might be useful for designers trying to discern some sort of pattern in this area of increasing complexity to carefully examine both changes in users that we are designing for, and changes in our capabilities of making things. This broad category can be divided into other dimensions, but for the purpose of this morning's discussion, I would like to talk about it in terms of the way users and the products of design are changing. We have become used to dividing products along a scale, from high quality, high price, to low quality, low price, and the normal model of course is that large mass products. Of course the structure of the model is relatively new to the world, and is a product of the industrial revolution. Mass society did not exist before mass production.

In 1960, if you knew the car a person drove, you could accurately predict the area of the town they lived in, the kind of house they lived in, the kind of school their kids went to, the kind of clothes they bought, and the kind of vacations they took. The consumer's behavior was relatively predictable. The hitch is, that it is not predictable anymore, the structure is fractured, and it is fractured by areas of special interest. What this means is, that there is an area of products and services which one can call good enough. This is where the individual wants the products to be utilitarian, but does not use it as a lifestyle product or as a product with which he or she has a high amount of personal identity. The entry of the early Japanese cars
was probably the first example of utilitarian products in this country where we see people wanting a cheap, efficient means of transportation. Status was not an issue. We see areas of growth in these good enough, acceptable, utilitarian products, while in contrast, we also see individuals and families not accepting those average utilitarian products, but going after extremely high quality things. We all know people with an interest in bicycle riding, for example, who would buy a bicycle for over $1,000, while the other aspects of their consumer behavior do not involve buying top of the line products. There are people who spend lots of money on kitchen gadgets, lots of time taking cooking classes and lots of time preparing food, and at the same time we see this growth in relatively high quality instant food and fast food restaurants. The same person buying the bicycle is not buying, the $50,000 kitchen. We see a wide variety of special interest areas, and within anyone of those relatively narrow areas of interest, the consumer generally believes that they cannot get anything that is truly good enough. When something new becomes available in that person's special interest area that individual has to have that new thing. Increase in performance is the primary concern, and price is of little influence. Thus, there is a fracture in consumer behavior with regards to special interest and utilitarian products.

Steve Wilcox: When you talk about consumer behavior being more difficult to predict, isn't that, to some extent, merely a function of what the predictive capabilities of a predictor are? Maybe, the models of human behavior just have not caught up with the types of patterns that we are seeing.

Whitney: Yes, I agree to some extent, although if we use the same methods of predicting that are now available and project backwards in time 30 years, and look at how people bought products or services, we would see the same clusters of behavior existing then, that we are now able to see using the new more refined
glasses that we are able to look through. For example, companies like IRI, pay consumers in order to monitor their purchases at the grocery check out, and sends out different ads in the mail to these individuals homes in order to see the affects that a particular ad will have when compared with other people in the same town who have received different advertisements. This is an example of what you are talking about and it allows us to see the pattern of consumer behavior more clearly.

Jack Nasar: I have a question about the model. There is a report I read a little bit ago, about a values project, conducted at Stanford, where researchers interviewed 100,000 people. As I recall, they identified some nine groups of people that had very systematic patterns of purchases across levels, from homes to cars, to the way they decorated their place. The study showed that there are, in fact, correlates within groups. It is more complex than the original two or three part models, but there is still some systematic pattern of predictability, and that differs from what you are presenting here.

Whitney: You are absolutely right. The model I have been describing is a simplification. In fact, there are patterns that can be discerned, and there are patterns more complex than the present model suggests. In fact it is not clear that the patterns of consumer behavior are not breaking down even more than either the Bales or Yankelovich et. al. reports indicate. The present model is a simple snapshot of the decomposition of mass society, decomposition that does not seem to have stopped with what Bales wrote about a few years ago. In fact, with the Yankelovich study, there appears to be a similar level of complexity, although the two studies differ in several respects.

The other issue regarding the question of complexity, involves the coming together of three entities: computation, communication, and production processes. In a sense, where they come together is where the action is, and in a
sense, this activity is developing a new medium. A new medium in a sense that photography was a new medium, a new medium in the sense that film was a new medium, a new medium in the sense that printing was a new medium, and we do know something about new media. That is, when they emerge, we do not know what they are, and we do not know how they work. The new media that have been developed in general, copy the forms and the conventions of the media they are trying to replace or improve upon.

Clive Dilnot: Printing may be a case of a medium that differs enough from its predecessor that it does not simply follow earlier conventions. Gutenberg first introduced the printing press in the late 1440's and within 50 years there were printers who understood printing, who understood it, utterly and totally from a typographic point of view. Given, that we are talking about new methods, or if you like, a spread of the technology, this is an extraordinarily fast development. It strikes me, that printing is one of the most quickly assimilated new media, in the sense of people thinking out its potentiality. In fact the model of the slow understanding does not even apply in this case. I think this is an important distinction that may have occurred partly because printing, in an odd way, was so different that one could clearly see the difference. This seems to be a case where the understanding developed at an extraordinarily fast rate. So fast in fact, that there have been very few developments in typography, or book design, since then right up into the end of the 19th century. Thus, in a sense you got out by 1500 the essentials, and there is little further development in the medium during the next 300 years.

Linda Ettinger: I think that example of printing is a good one. If we look today, however, at the industry of printing per se, in light of the electronic and information age, I think that we would have to say that there has been very little
change occurring in terms of the potential of how things can be reconfigured. Perhaps that is because we are still living very, very tightly within the boundaries of printing as it is defined through history. I think that design, in terms of what you are trying to get at with these models, needs to be concerned today with boundaries, with the legislative perimeters that currently exist because of the ways we have conceptualized and constituted our world at this point. Printing is a wonderful example of an area that cannot change in light of the information age right now because it is stuck in a design boundary, based on not just history, but on language, systems and categories.

Marcos Novak: I think a new medium which is a contemporary of the present, for which we can not seem to find a task, is the kind of situation that is hard to call a medium, even though it concerns what will eventually become a medium. Any kind of situation that we would call a medium is a situation which reacts to us in an unpredictable way. We can see it initially in art where there are sculptures to which the person does one thing and it does something unpredictable. We do not know exactly what it will do, but, what one does affects what the medium gives us as a message.

Whitney: I think what you described is what we call human conversation.

Novak: Well, it need not be exclusively human.

Whitney: I realize that you are not talking about human conversation, per se, but about a mode of interaction, where we are interacting with a set of information, which has in the past constituted interacting with other people. The lump of information, as it were, is normally contained in another human being and the resulting interaction with that information by means of conversation, can be used as a metaphor for how we interact with a variety of things.
Novak: Well, if you twist the concept that far, you could say that printing itself is, a monologue, it is just human conversation of a certain sort. The book itself then, is just a monologue, so it is also a human conversation, but I suggest that perhaps that definition of a conversation is too broad.

Whitney: Printing dramatically modified several things, for example, our dependence upon memory and the nature of storytelling, which is a one way human dialogue. Also, an important aspect of printing as a medium, that cannot be reduced to merely an issue of the grammar of the medium, is how the medium gets used and who uses it, and who has the availability to it. Although the grammar of typography may have been described and defined very quickly, who used printing and who used books and how it affected society and how that in turn, affected printing occurred over a much longer time frame.

Returning to the discussion of consumer behavior, the first models concerned society and product types. It is possible to take this scale of high/low quality, and repeat it, using designers and design fields as a variable rather than consumers and product type. Traditionally, there have been a small number of designers who are designing very high quality items and a larger group of designers interpreting other kinds of needs. My composition is, that complexity is adding a new dimension. There are many more designers today focusing on more complex issues than ever before. For example, corporate identity programs; how does one get a corporate identity for something like Beatrice Foods? They sell everything from stereo equipment to skis, from hand dryers that go in washrooms to food, and some of the food is high quality, and some of it is low quality. How does one go about describing the more complex problems? It is possible to find design firms that focused on being able to deal with large complex problems, but the hitch is, that many designers tend not to view those things as very exciting because the complex
problems do not follow traditional models. I think it is really a challenge for designers to see what is up there. How do we deal with the complex problems at the level of quality that we have become used to in the tradition of our field?

Raymonde Guidon: I would like to have a more precise definition of a complex system. What realms of our experience are becoming more complex? Your claim is that this new dimension of complexity results in part, from social changes; can you explain that?

Whitney: Sure, we do not know who we are designing for anymore, and to further add to the complexity, we do not know how to make things anymore, we do not know, for example, if a product should be smart or dumb.

Ettinger: I am thinking about your model and the issue of complexity; another way to think about it may be that perhaps, the problem of complexity wasn't just that things were easier to describe in the past, because there was less to describe and the audiences were more limited, but the problem might be that we thought things were less complex in the past than they actually are. I think the problem might be restated as: who is in control of the design process, and who is doing the designing for whom and what are the invisible issues in design that aren't being addressed? How do we find those issues, and what do we do about them? I really like the statement by Mihai that, "design is determined by the pragmatic context," and I would say that your model might be leading to modification of that statement, to one that suggests that design is the pragmatic context, and not merely determined by it.

Whitney: To some extent, I agree, but this model is offered simply as a way of illuminating some of the issues concerning the concept of complexity and its relationship to design; the model itself is not what is important.
Ettinger: Well, it is the notion of complexity I am trying to get at. I am not real sure what you are talking about.

Whitney: I can understand some confusion regarding the concept of complexity, the notion of complexity is certainly a difficult concept to nail down in precise terms. However, since this is probably the most theoretical design conference ever held, perhaps we will have the opportuniy to arrive at a better understanding of the concept.

John Rheinfrank: I would like to offer an example which might help to clarify the issue surrounding complexity. Representatives from General Electric were in our office the other day, and the statement that they made to us was that since they combined with RCA, they now had a hybrid of two distinct cultures which would have seemed completely incompatible before the merger. They also acknowledged that they now have an array of technical possibilities that are essentially unlimited and that they did not know what to do with the potential. They had plenty of ideas flying around regarding the individual capabilities of the two companies before the merger, but these ideas seem to have no point of integration which would allow for the realization of their new potential. They came to our firm and said, “well you have been successful in the past in dealing with complexity of a similar nature. For example, Xerox was a failing copier manufacture, and the same kind of confusion existed there. So, can you draw upon your experience in dealing with complexity and come up with tools that can help General Electric achieve an integration of its combined potential, from what now appears to be a confusion.” This so called confusion might be what you (Patrick Whitney) have been referring to as complexity, and at this point I would have to disagree with you a bit, for I think complexity is a good thing. I think we simply need the tools, the organizational
skills, to deal with it a little more successfully, and the inherent benefits in this sort of complexity will become apparent.

Richard Gorski: I would like to add to what John Rheinfrank said, I think that one of the problems of looking at models like the ones proposed by Patrick Whitney, is that we are looking at a duality and I think we have to look instead at relationships. I would call it a false dichotomy between what was described earlier as a distinction between old means of manufacturing and new, call it, flexible means of manufacturing. Now if one sees this as not dualities, but as false dualities, or false dichotomies and then one sees instead of an either/or, a both/and, products that both have a regional specificity and some form of commonality. Perhaps this is a different notion of commonality, something other than the idea that everything is the same for everyone, which is a trap stemming from the false dichotomy.

Klaus Krippendorf: I wonder if this notion of complexity is not perhaps misleading, because as John Rheinfrank has said, complexities disappear as soon as there are tools to cope with them, or concepts to describe them. Complexity is actually a projection of our inability to cope with something, and if you describe things in terms of its being a complexity, one does not really get at the problem. I think we have to focus on uncertainty and describe things using that sort of descriptive vocabulary.

Rheinfrank: Complexity is really a conceptual framework which prevents one from ever attempting to address the problem.

Krippendorf: That’s right that’s what I wanted to say.

Whitney: I think what you have both done is introduce the next thing that I want to talk about, which is describing the tool, the organization structures we have at our
disposal for dealing with complexity. But first, in reaction to what has been said, we do call things complex when we don’t understand them, but also, clearly, it seems to me, there is something to the notion of complexity that is distinct from our level of understanding of any particular problem. Clearly, some things have more parts than others, and the structure of some patterns of behavior are more nested than others.

Nadin: No, no, I absolutely disagree. The designer deals with complexity in order to dissolve it for the user. I do not want to transfer the complexity to the user, but I myself must deal with complexity, which means what? I am becoming more and more aware of all those numerous parameters involved in something that I am going to design. I can say from the beginning, this is not a complex problem, I am going to ignore x, y, z and I will concentrate only on 2 or 3 variables. But, if I as a designer, in a conscious decision say, “I want to deal with as many significant parameters and their relationships as possible, then I’m going to work in complexity and in the complexity of design. That is why I really believe that Linda Ettinger’s question was very correct. Complexity can be defined very precisely. If I have a design problem, my first task is to work on a cognitive level in order to identify the parameters which are essential to the solution. Next, so to say, we work down through the parameters in order to see the importance of the first, second and third elements. In order to see what can be ignored, what cannot be ignored without a certain price. It’s part of the trade off which makes design what it is.

Whitney: I do not think what you have said really contradicts my statement.

Nadin: I did not contradict you, I only did not want to accept the idea that complexity is not an issue, it is a very important issue if we look at complexity not as
the result of what we do, but rather, as the angle from which we look at the problem.

Whitney: Well, then you would agree that certain design problems will be viewed as more complex than others. The corporate identity program for Beatrice, for example, is more complex than a corporate identity program for some other less diversified corporation.

Nadin: This, I do not know, I would say for Beatrice that. I have a very good corporate identity program, I would take a Mickey Mouse and take a name and attach it to Beatrice. For Beatrice, it is as much a part of normal development to sell computers and food, as it would be for me to consider all of the marketing happenings that made Beatrice to be what it is today. I would say if you want a good corporate identity program, stick with a Mickey Mouse for the next 20 years and everyone will know it is the Mickey Mouse that is speaking.

Whitney: That would not work for their stereo equipment.

Nadin: It would, it would.

Krippendorf: I would say that complexity does not exist in the world, it is a property of cognitive activities of the designer. It does not really exist and one can choose to ignore it. In fact, that is what you have been describing, if one thinks there are very many parameters and that all of them have to be considered, then the problem will be complex and confusing. This however, is really a problem with the designer, with the inadequacy of his or her conceptual framework, or with the cognition involved. Complexity cannot be measured in reality, it is only the designer's conception that can be measured.
Whitney: In a pragmatic sense, there are projects, which for all intensive purposes, when considering the success of the project, are more complex than other projects.

Krippendorf: If you choose to see it that way, yes.

Whitney: Well, if you want the project to be a success within the context of its being useful to people, then some projects must be viewed in reality as being more complex.

Krippendorf: But who establishes the context? You?

Whitney: No, that is where I disagree. We do not define the problem, the problem is defined externally by several forces, we are one of the actors, but surely we are not the primary force controlling the shape and complexity of the problem.

Novak: I would like to argue a little bit for complexity, because it seems to me that one does not always want to hide it. A little quote from Einstein comes to mind, he says, "Everything should be made as simple as possible, but no simpler". Some things we just want to be complicated. If our problems are becoming more complex, we need to know, more complex than what? Is the corporate identity system, or the functioning of the television, or the thing that we send up to space more complex than an art nouveau building? Is it more complex than Mozart's music or Beethoven's symphonies? I think that we have different kinds of complexity. There is a complexity which is comprised of simple parts, or of simple components, and there is a kind of complexity which has to do with communicating meaning to people, which is enriching and which we would want to put into the project. In addition there is a type of complexity that involves doing what Mihai was suggesting, namely taking a problem and making it as complicated as we can for ourselves, before we solve it, because that is how we make it better. Before we
say that we want everything to be simple, I think we should say that we want everything to be rich, and if it is simple, that it is good, but if it has to be complicated, well, that is what it has to be.

Steven Wilcox: I find it interesting that we have stumbled across a very fundamental logical issue in our discussion of complexity, and that fundamental question concerns the very nature of knowledge. The question could be formulated in a number of different ways, but in the present discussion we find it emerging from our inquiry into the locus of complexity. The first position is that complexity exists in the world for us to sort of stumble upon, some things are complex, others are not. That is a dissatisfying answer, so the next position is the rationalist perspective, which claims, in contrast, that complexity exists in us. There is no complexity out there, the world is just this undifferentiated mass we impose the complexity on. I would like to propose that both of these positions are fundamentally wrong, and that in reality, complexity, just like everything else, comes out of a relationship between the knower and the known, or between the organism and the environment. Philosophers like John Dewey, back in the 1920's described this sort of phenomenon and explained very clearly why both of these kinds of traditional views are incorrect. He went on to suggest, that at all times one needs to consider both the framework from which something is looked at, and what it is that one is looking at. One cannot talk about either one independently, they always have to go together.

Whitney: Absolutely.

Krippendorf: I am surprised that you say absolutely, because I say, absolutely not. The reason is very simple. How can a law describe what is known without the person knowing it? There is nothing independent of what one knows. Either one
knows it or one does not. A person can not describe something to be known and what he or she knows separately. Everyone lives in his or her own world that has been created by the cognitive structure that each individual imposes upon their own experiences. One cannot make assertions about what is known independently of what one knows. It is, in fact, during the last 20 or 30 years that cognitive theories have become precise enough to reveal the false ontology that is created when we conceive of the world as existing independently of us knowing it.

Wilcox: I agree with you to the extent that you claim that one cannot describe anything except from one's own point of view. However, I think that this sort of radical rationalism that you are suggesting fails to capture the idea that if I turn around, I discover something. The concept of discovery is left out; I do not create what is behind me. I can only see it from my own point of view, but it still exists independently of me in a very basic sense, and that is what traditional rationalization, similar to what you are advocating, misses.

Krippendorf: I don't think so, I do not think a person can point out something that he or she has not seen.

Ron MacNeil: What does the viewer that is observing the scene from a window see? What does a person looking at complexity see? We are trying to discern what is in that person's head and what is in the environment.

Jack Nasar: From an empirical perspective, I would agree with what Steve Wilcox has said, namely that it is, in fact the transaction between the individual and his or her environment that defines the ontological relationship between the knower and what is known. Based upon some evidence that I and a number of people have gathered, it is apparent that an objective definition of the features of ones surroundings can be achieved. Specifically, with regard to the issue of complexity,
it is possible to have a number of observers respond to a particular situation in terms of its complexity and get consistent responses. That is to say, the observers will consistently call one thing complex and another thing simple. This is not to say that it is all in the environment, obviously there are going to be some individual differences across people, and obviously the interaction of the individual with his or her surroundings is important. Nevertheless, it must be recognized that there is something out there in the world to which we respond. We look out the window and we probably would all agree that there are trees out there and that what they constitute is a fairly complex visual scene. It is a shared perception.

Krippendorf: How can you share perceptions? You can share only words.

Nasar: It is possible to share behaviors in response to complexity. It is possible to put playground equipment outside for children to play on that varies in complexity and observe systematic changes in behavior that is preserved across the sample of a child being observed.

Krippendorf: Well, you say that is a tree, and I say it is a tree and we agree. But it does not mean that you see the same tree that I see. We cannot share these experiences, except through the same language that we use. In fact, many of our observations are qualified by the words that we use to describe them.

Whitney: In a particular setting, people sharing a common language and culture seem to act in a similar fashion and go about solving their problems in similar ways. The patterns of behavior are similar enough that someone might design a product, put it on the market, and later observe that everyone uses it in much the same way, and not be surprised by this observation. The behaviors are similar enough to allow someone to design, for example, a typewriter, and to be able to anticipate how that item will be used by the people that come in contact with it.
Let's say that complexity is neither good nor bad, but rather that projects seem to be getting more complex or at least designers seem to be getting more confused. If we look at what design schools are teaching and at the way design offices are practicing, it is fractured. Is there something that we could offer designers that would help them be less confused? Is there something else that would help us to be less mystified by what we are facing.

What we have been given of course, is a new tool called computing, which when merged with production and communication systems can tie them, to some extent, into something that I would like to call the information utility. The information utility contains two major categories: knowledge and abilities. Knowledge refers to concepts, and to the relationships between concepts, and abilities refers to the manipulation of that knowledge and how it is represented. Under the category of knowledge, we have what can be called general knowledge of the field or domain. The domain may be product design, architecture design, graphic design and so forth, but at any rate it consists of general knowledge of the design field. In contrast to the general domain knowledge, there is project knowledge which is in principle new information that does not exist in the general fields knowledge. It is specific to the project upon which the designer is working. In addition to general domain knowledge, and specific knowledge, it might also be desirable if the information utility contained knowledge that pertains specifically to the designer using the system.

The other category involves what might be called abilities, or in other words, what we would like the ideal system to be able to do. The system should have the ability to gather information from a variety of data bases, it should have the ability to store information in an organized format, it should have the ability to notice and suggest patterns or relationships that exist between the stored information, and perhaps the most interesting aspect is that the system should have the ability to
transform or create new information. This ability to transform information may involve a manipulation of the information that influences the presentation of that information and the form that it takes in any of the various sign systems, including symbol strings, diagrams, and pictorial representations. However, in terms of the potential utility of this system for designers, the transformation within the system would be most useful if it were expressed as a simulation. The simulation would allow the designer to watch the proposed products being used in various situations by the proposed users. It would be like watching a movie, except instead of putting real things and people in front of the camera to get pictures, these images would be completely computer generated. The parameter necessary for the computer generated interaction would be derived from information about the user, which I have referred to elsewhere as the physiological-cognitive-social value human profile, (Whitney, 1986) and technical information about the product. The technical information about the product would include such things as the characteristic of the production material and the production processes, as well as specific project information, such as the competition of the company that the designer is designing for, product history, distribution channels and whatever other information that normally enters into making a design decision.

Editor's Note: In order to generate examples of what an expert system, or intelligent design machine might be like, including the desired knowledge base and abilities to be possessed by such a system, and how the concept of an information utility might be applied to various design fields, the symposium participants were divided into several small groups. Each group was asked to define a particular design field, such as product design, architectural design, software design, etc., a particular design project, such as a new keyboard, a corporate identity program, a new school, etc., and a particular user type, referring specifically to the intended
user of the proposed expert system. Each group attempted to provide as specific an example as possible of an intelligent design machine that would facilitate the design process for the defined design field, project and user. What follows are excerpts from a presentation describing some of the desired specifications of an expert system to aid in the design of a total banking environment. The total banking environment was described as including everything from the design of the physical layout of the bank interior, to the message content of the bank's brochures.

Liz Sanders: The design field, in which the proposed expert system is intended to operate is defined as the area of information design. Information design can be thought of as involving creation and the manipulation of any potential source of user information: from the structured array of symbols on a piece of paper to the geometric structure of surfaces and substances in the environment.

We picked as the design project a prototype bank environment, not just a single environment, but a system of environments. The prototype bank environment would be one that could be modified to fit different geographical regions and different demographic patterns. The client, we decided, wanted us to address both the structural and symbolic components of the bank environment. In order to address this broad issue, we chose to define everything in the environment as a potential source of information. In other words, the interior layout of three-dimensional surfaces, in addition to the words and images on two-dimensional surfaces were viewed as providers of information. We chose as a user of the expert system, the designer of the modular banking environment.

What we did next, was to describe the knowledge base that we would like to have included in the expert system, as well as the various ways we would like to see that information manipulated. Under the heading of general knowledge, we thought it would be useful if the system possessed knowledge of human spatial
behavior; more specifically the stereotypical patterns of interactions within various behavioral settings and how those interactions are influenced by different physical layouts of surfaces in the environment. Knowledge of patterns of human interaction with the information content represented in various two-dimensional forms would also be included within the system. Another component of general knowledge that we thought should be possessed by the expert system, is knowledge of the elements, or primitives that the designer would be working with, such as three-dimensional forms, colors, textures, and both visual and verbal representational components. Knowledge of the various tools and techniques available for manipulating those elements would also be desired.

Under the heading of specific project knowledge, we wanted the system to know things about the users for whom the operator of the expert system would be designing. A detailed human factors profile of both the bank customers and bank employees specifying their abilities and inherent limitations, on both physical and cognitive levels, as well as a profile in pertinent social values should be included. The system would also need to know something about banking, namely, the type and nature of the various transactions to be anticipated, as well as specific security issues and the kinds of transactions to be avoided, or discouraged. Knowledge of the particular client bank would also be needed, including its various philosophies and projected strategies with regard to banking, as well as that bank’s plans for the future. An understanding of the bank’s competition, and the competition’s future plans, to the extent that such things could be known would also be useful information to include in the knowledge base of the expert system.

The user of the expert system, in our case, would be the person designing the bank environment. Since, we conceive of this expert system as an intelligent tool capable of aiding the designer without inhibiting or impeding the design process, we thought it would be important for the system to know about the designer’s
capabilities. What kind of designer they are, how they approach a particular problem, and how good they are at approaching the problems from various perspectives. We thought the system should also understand, or know something about that designer’s design philosophy, as well as their understanding and learning mode.

In turning our attention to the abilities of this hypothetical design machine, we considered it a tremendous asset for the system to be able to generate new knowledge structures from information contained within the system. This generative process could take the form of a simulation. Derived from a sufficiently rich knowledge base that pertains to the user of the product, and the design elements that are being manipulated in the formulation of the product, the system could simulate the interaction between various users and the various facets of the products being designed. From the simulation, it might be possible to detect patterns of interaction between the numerous levels of the complex product being designed and between the product and the user of that product. These patterns might then, in turn, be used to generate new forms, new rules of formulation, and new restrictions or constraints that would limit undesirable interaction.

We believe the system should be able to store and retrieve information at various levels of details within the nested structure of relations, and use various modes of representation and various media for the presentation of the information. The system should also be able to notice patterns and make suggestions to the user on the basis of vast network of link-relational data, either throughout the design process, or upon demand. The user could ask to be reminded of certain key relationships and subgoals, or interrupted if specific design principles are violated. With regard to the system’s ability to convey the design solution, we felt that it needed to not only simulate the physical components of the bank’s layout and the various information pieces, but that it also needed to describe
some sort of document or guideline by which the various elements could be combined in a variety of ways to fit into the various locations. One way we have for explaining our work at RichardsonSmith is to think of a design language, and for this hypothetical design project we would probably desire a similar solution and describe it in a similar way. We would not design every single bank, we would design the pieces and then the rules for combining the pieces into coherent, well formed wholes that fit the users and the geographical area. So, the end result would be products, a representative sample of products, and processes and interaction styles along with a document that described the philosophy behind the design and how the pieces fit together in future applications.

Nadin: In other words, we looked at a project not as just an intelligent package, but how a machine is contained in a machine. We are talking about design machines that will create and provide information.
PART III

Design Language

Discussion led by John Rheinfrank
Rheinfrank: I would like to share some of our work with all of you because I feel it is an opportunity to provide an example of the interplay between some of the theoretical issues we have been discussing and the practical application of the principles that are at the heart of that discussion. I have this morning a few examples of some of the projects that we have worked on that I feel might contribute to the knowledge base.

We might take a step back by first asking, "is there a knowledge base there to which one might contribute?" If we answer yes, as I think we have done at RichardsonSmith, what knowledge in particular will contribute to our better design? To put what I am saying in the context of this morning’s discussion, we have found a particularly useful tool for dealing with the issue of complexity. We call this tool a design language. A common sense way to understand what is meant by a design language is to think of the example of the banking environment that Liz Sanders presented. The banking institution and the environment it creates has a composite of ways in which it expresses itself to its clients and to its employers. What we are doing is simply designing the composite expression through the notion of a design language. In other words, we are trying to identify the components of the problem, and help the client develop a design language that can be used to resolve various complex issues.

In order to give an example of what a design language is, I will review a project that we did for Xerox over the the last five years.

As a side comment, I would like to mention that Xerox is a company that seems willing to take the risk of supporting research. They have been a wonderful client, allowing us essentially unlimited time and resources to find out what they should do with no more restrictions on us than that. I must also say that this work I am presenting does not come from one person, it represents the contribution of between 20-40 people at RichardsonSmith.
Xerox came to us with a problem, they had lost market share, and had in their sites the eventual failure of the company. They recognized as their most serious failure the fact that many of the users of their machines were having difficulty understanding how to operate them. The machines are complicated, and in all likelihood, with recent technological advances, will be more complicated in the future. Xerox claimed to be able to see the day when they would have to put a users manual 3/4 meters thick on the machine in order for someone to be able to operate it. The challenge at RichardsonSmith was to tell Xerox how to design a machine that does not need a users manual. This task of designing a machine that does not require an operating manual has no doubt been accomplished for very simple products, but never before for something as complicated as the class of machines being produced by Xerox. What is essentially being called for is a machine that has the ability to inform the user, by using both the design of the physical structure and computer graphics to display information. This view of machine as provider of information has come to be a design language, or in other words, a way that a product can express itself to its users through the form, the shape, the color, the graphics, the content of an alpha-numeric display, and diagram on the screen. In essence, what the machine is and how it is used, can be seen by the user, and the product provides that information.

In a sense, we were dealing with growing complexity. We had to anticipate all of that complexity and come up with a language that could be used to embrace it. We found that this design language can conveniently be divided into 4 domains. A possible fifth domain called the signature domain is particularly relevant to many clients, but it is an area that has traditionally received a lot of attention. So, I'll devote most of this morning's presentation to some of the more novel aspects of the design language, and describe these aspects using examples derived from the project we did for Xerox.
The first domain, involves what is called the task domain, and in very common sense terms, focuses on what the user is doing, and on what he or she is attempting to do. As an example, it is possible to think of a copier as a mysterious box into which the user puts an original, and out of which he or she receives a set of copies that corresponds to some image that they have of the results. Focusing on the idea of a task domain, it is possible to convert the idea of copier as mysterious box into the idea of copier as workstation, or as a place with stations of work that one moves through as they make copies. This latter idea makes use of a model of the world that many people have, a model that can easily be transferred to the copier machine. It is a very simple idea, but very powerful to the users. We designed the copies so as to be compatible with the users model of the world. We designed the copier to appear and to function as workbenches or tabletops, instead of as mysterious boxes that sit pretentiously in the user's environment. This model of a workbench is enhanced by design features that clearly demarcate the areas for a particular type of interaction. For example, light colored surfaces are used to designate those areas related directly to the activity of copying, while more darkly shaded areas are used to designate places that the user can enter the machine, to load paper, to clear a jam, or to service the machine. We assumed that the user has a comfortable relationship with the other things in the office; the other tabletops or workbenches, so we used the metaphor of the tabletop as a design tool, and designed the copier in a way that would support that metaphor.

The semantic domain involves designing areas that contain meaning. In other words, instead of using color in the traditional way which is to make the machine look as unobtrusive as possible, colors were chosen that convey information to the user about the workings of the machine. The colors were manipulated in a systematic fashion, and the system of color coding was maintained from product to product. For example, surfaces that are blue designate the place where the paper
supply is loaded, green signifies where the original is placed, or where the operation begins, and red is used to signify where the user picks up the copy, or where the operation ends. We anticipated that the future machines may have as many as eight to ten different places to load originals, and an equal number of places to pick up the output, so some system of coding would be needed. We found that when we put uncoded machines in front of people, they would try to put originals in the output slots and be waiting by input slots for their copies to come out. We found that color coding was a very convenient way for the machine to convey information about itself to the user.

The next domain is the syntactic domain which involved having expectations about where things are going to be. For example, on all of the products, the work always flows from right to left, so that the user knows, in every instance, when they walk up to a new machine that it is going to have a right to left work flow, and that their point of focus starts at the right and ends at the left. This might at first seem to contradict with the users expectations, for we tend to describe most things as flowing from left to right, a stereotypical flow that is consistent with the direction in which we read and write. However, what is even more critical from a human factors perspective is that things have to be positioned with the right hand and because of the kind of machine that this is, the flow of work moves naturally from right to left.

The engagement domain is a domain that has to do with motivation. In other words, what is it that we can do in the design of the machine to make it more engaging. I don’t want to use the word fun, because I think the word never applies to making copies of somethings, but what can we do to make the user not mind being there? One way that we tried to achieve this goal was to include an animated mimic diagram that depicted the way paper goes around inside the machine. In addition to giving the machine some movement and life that the users find
engaging, it also maps a series of 7 or 8 locations within the machine that paper can get jammed. The mimic diagram saves the user from having to get that information out of a printed instruction manual. The mimic diagram simply flashes the location of the paper in the machine by there being a one-to-one correspondence between the location of the paper on the diagram and its location in the machine.

Having briefly examined the domains of a design language, it is now possible to take a closer look; to examine what we are dealing with inside this collection of design domains. At this level of analysis we find that a design language can be divided into six segments: primitives, a description of the primitives, a grammar, a language of the language, a concrete demonstration of the language, and finally, a set of arguments that support the decisions being made. At the finest level we have the design primitive, for example the use of a green arrow in the registration corner to point out to the user where the original is to be placed.

The description of the primitive, in the case of Xerox, would be a printed document that describes all of the machine primitives that have been employed. The description of the primitives need not, in all instances, take the form of a printed document, we can often merely observe what the primitives are and describe them to each other.

The grammar describes the way we combine the primitives into meaningful wholes. We can think of them as well formed sentences; they are well-formed wholes, or composites of primitives within the machine’s surfaces that the user can make sense of.

Wilcox: Can you give me an example of what you mean by primitives?

Rheinfrank: The area on the copier machine into which the original is placed makes use of several primitives in our way of thinking. It has a shape primitive, a cavity or concave surface which, in general, specifies a place where something can be put,
and, in this instance, it specifies the place where the user is to put the original. We have also made use of texture primitives with recessed textures being used to specify the areas where the user puts things in, and a raised texture to specify a place where the user gets things out. Color is also used as a primitive, as mentioned earlier. So these are a few examples of primitives, and then there are rules that specify the way the primitives can be combined.

Wilcox: If you provide primitives and you provide some rules for their combination, isn't it true that there must be rules for what constitutes an illegal combination of primitives?

Rheinfrank: When one is combining primitives there is of course the possibility of getting some illegal combinations. These are usually combinations which confuse or convey the wrong meaning. To give an example, if the designer were to design input devices that had negative spaces, he or she would not want to violate that primitive by putting a big bump in the center of it which, on the whole, would create confusion. This type of combination of primitives would be outlawed.

While the set of primitives is limited, the system for combining primitives is, however, essentially open. If we think of the grammar as generative, that means it should be open-ended, and that one should be able to substantiate any number of combinations of primitives in much the same way as one uses natural language. What we have also allowed for, with this system, is an ability to move from generation to generation in a line of products while maintaining the same design language.

Wilcox: Can you give me a more concrete example of a design language?

Rheinfrank: We do not have enough examples from our experience to make it any more concrete. Right now, a design language is just a set of common sense
agreements between Liz Sanders, the other people who have worked on various projects and myself. However, it seems to be something into which another designer can enter without any preparation and find out what it is all about and how it operates. Don’t force me to describe it in more precise terms.

Since the design language is often difficult to express in words, we found it very important for our client’s sake to construct a series of demonstrations of the design product, models which capture the design language. They are not real products; they are products that imagine future technologies and users and respond to those technologies and users with real proposals—things built out of wood and Plexiglass and things that behave. What is produced are not mere images and words but models that behave and respond to the people who use them.

Ettinger: Can people design their own models, or can they only edit them?

Rheinfrank: They can edit, or they can move outside them. In the case of a later generation model, moving outside of it is often a point of discontinuity rather than planned continuity, but, in general, both of these things can be done.

David Snediker: Xerox has obviously made a corporate decision to go the route of repairability as opposed to reliability. That was a given presented to you as designers, was it not?

Rheinfrank: No, actually we found a group within Xerox who were interested in customer service, and they had produced a number of reports that said that Xerox could save billions of dollars over the next several years by becoming involved in customer service. It was all still on paper; they didn’t have any idea of how to enable that. So what we did was build models and apply the principles that we have been developing to the notion of customer service. The conclusion from our analysis was that Xerox had already achieved a very high degree of reliability, and
that to achieve the next percentage point, or the fourth decimal point in physical reliability, would require a tremendous investment in dollars. Nevertheless, they had the engineering mentality and were driven toward the ideal .999999 in reliability. What we were able to demonstrate to them, using a series of experiments, was that what really matters is the perception of reliability. In other words, a particular machine can fail as many as 27 times a day and still be perceived as being as reliable as one that fails only three times a day. It is easy to imagine the savings in terms of development costs that can be achieved with a machine that needs to be only one ninth as reliable as some other machine.

Snediker: Is this difference a perceptual difference or a functional difference in terms of the availability of the machine?

Rheinfrank: Reliability is measured in terms of the availability of the machine, the time it takes to recover from the machine failure and the user's ability to successfully carry out the recovery task.

Snediker: So, it is not related necessarily to the mere appearance of reliability, or, in other words, the belief that because the machine has a neat, clean design it must be reliable?

Rheinfrank: No, I am referring to the machine's actual reliability, or its availability for use.

Snediker: One interesting thing with regard to assessing a machine's reliability, from the user's perspective, is that if people are invited to try to find out what is wrong, and if they are successful in fixing it, they typically do not consider it as an instance of machine failure. However, each time the person walks up to that machine and finds that it is unavailable for service, it is counted as a failure, and it is
counted as an additional failure every time the person returns to the machine and finds it broken down, even though he may have gone to it three times during a single hour. So, if the user is able to quickly fix the machine, interestingly enough, it is not considered an instance of machine failure; the notion of a failure disappears.

Cavalier: Can you safely come to the conclusion then, that it is not a matter of whether a product works or not, but that it is a matter of how you frame it? In a sense, what I see you doing with this equipment is coming up with alternate ways of framing it, and framing on any number of different levels.

Rheinfrank: Yes, in fact Shank and Ableson's notion of a frame was conceptually one of the very early influences on this notion of a design language.

Hans van Dijk: John, it is not just a conceptual problem, I hope, but it is also functional; otherwise, the product will end up like Mussolini's highways; everything looks terrific, but they fail eventually because they simply do not work.

Rheinfrank: The new copiers have very few actual improvements in function, and, in fact, in a great many cases, Xerox found that they could degrade the function without impairing the reliability; so, it is not a fascist notion, or the mere appearance of reliability.

Nadin: Well, we all know that language is a fascist phenomenon.

Snediker: John, I would like to point out some important factors influencing the results of the experiment on copier reliability that might not be apparent from a simple comparison of copier failures. The number of machine failures represents a negative value, or what the manufacturer can get away with, but it does not reflect the positive attributes of the machine that allows the negative qualities to be
dismissed. One reason that Japanese competition is succeeding in the market with copiers of inferior copying quality that break down 30 times as often is that the designers there have grasped what might be called a “hands-on-architecture.” In other words, the machine may be a monster and it won’t run; only the user knows how to fix it, and that seems to be what counts.

Rheinfrank: Yes, I would agree that that is an important factor influencing the user’s perception of the machine’s reliability.

We found that it was crucial for our clients using the language to be provided with the rationale for using it in the form of principles, as well as a rhetoric that argues for why the language is a good thing. Our best capturing of that argument was a matrix which listed all the features of the language horizontally and all the business services vertically. The cells in the matrix were scaled, with the big cells indicating a major benefit, the empty cells reflecting almost no benefit and the intermediate size cells indicating some moderate amount of benefit. So the user of the language could go into that matrix, identify a point and then create an argument around that point.

Shifting focus slightly, I have another example of the application of a design language, a projected office system which is intended to replace the present use of panels in creating office cubicles. In this example, we are essentially designing the environment in which the products from the previous example (i.e. the Xerox copier) might appear. We are developing a language for furniture, enclosures and space that would be available in the workplace within approximately ten years. We found that the notion of impermanence and the negative connotations that it creates are frequently experienced when one is inside the current office panel systems with the desk and files hanging from the panels. Many companies are trying to improve the panel system. They listen to the list of complaints about
panels, pickup on a few things and perhaps add more soundproofing material to the panels or make them out of steel instead of wood so that they will perform better. We think this approach is basically flawed.

As an alternative, what we have done in conjunction with the architect, Christopher Alexander, is address the conceptual core of future offices; where we found notions of permanence and working together, instead of impermanence and isolation, to be at the heart of the matter. Our solution involved using the concepts of permanence and working together and applying them to the design of a future office space. As a result, the walls became thick instead of thin; they grew to over 7 feet from their previous stature that often failed to reach a height of 5 feet. The modules included such forbidden things as ceilings, and had cool light above and warm light below. They were very compact; in fact, they were about 30% smaller than existing offices, but had the perception of being as big or bigger than the existing cubicles. They had familiar things like base boards and crown trim along the top which is something that gives people a sense of space that they absolutely do not get from panel systems.

Ron MacNeil: Is this feeling of permanence merely illusionary?

Rheinfrank: No, the feeling of permanence that one gets comes from the fact that a person could lean on something and it would not move, like a real wall. In addition, a sense of permanence comes from the fact that when one looks through something he or she sees texture, and the views are broken or modulated by elements in the environment.

What Chris Alexander and I have been finding out is that his notion of pattern language and our notion of design language at RichardsonSmith are almost the same thing, and that instances of pattern language are very much like instances of design language as typified by the design language for Xerox. Again, the design
language is a way of dealing with this thing that we call complexity and, in our experience, a very powerful way of dealing with complexity.

Nadin: During John Rheinfrank's presentation I briefly referred to the fascist mentality of design. Obviously, the use of such a strong qualifier imposes the need for clarification. I know that, due to its very strong political connotation, the word "fascist" implies a value judgment, and I will not shy away from saying that it was a value judgment. In the context of our discussion the word "fascist" was used along the line of the analysis given by Roland Barthes. His perspective is one of language, more precisely, language as described in the structuralist approach of Ferdinand de Saussure. Barthes noticed that fascism can be seen as a totalitarian structure imposed upon a society. Its rules are not rational; its dynamics are one of delusion. The individual is negated; repression extends to the sphere of intimate life. Barthes suggested that there is an analogy between fascism and language in the sense that the characteristics of fascism are, to a certain degree, characteristics of language. He went on to say that we are born in a language we cannot choose, that the rules of the language are imposed upon us, that language represses individual expression, and that this repression extends far beyond the spoken and the written language. He looked at fashion, photography, and human communication and noted how the same characteristics apply to these. The participation of language in deception is perceived as the result of an accepted system and of shunned responsibility. In using language, people participate in their own deception and in the deception of others. This is not the same as saying that, because we use language, we are all liars or deceivers. Barthes was actually interested in the circumstances that make a fascist structure possible.

We take language for granted. We take so many things for granted- design, for instance, and we fail to question language when it no longer accomplishes the
function for which we use it. We also, it seems, fail to question design when a certain product is a definite failure. We use language without thinking about it too much. We design without considering the implications of design. The result is a kind of generalized irresponsibility. If everything works fine, we don’t look for reasons to question what we did. If things don’t work fine and they did not work fine with fascism which was a tragic political reality, we claim innocence. From “I did not know” to “I’m not guilty,” “I was not a part of it,” and “I actually never liked it,” everything can be heard from people who otherwise display a high sense of responsibility.

We do not invent our own language, but we still have a responsibility in how we use it. We usually don’t invent our design, but we have a responsibility in defining the concept that we apply when we design. There are several concepts: the morphological concept of design, dealing with form; the functional, dealing with the functioning of the designed object; the structuralist, concerned with the relation among the parts of the designed object and the relation between the designed object and the environment; and the postmodern, dealing with the recycling of design. If we are not aware of our choice, then all we do is follow a trend, be fashionable. There are many designers who will do no more than be fashionable. When designers are fashionable, they place themselves in an environment of imitation and lack of responsibility. We can easily write computer programs that will generate design by a certain formula.

Today, we are in a state where we become more and more aware of the power of language, of the power of semiotics. John Rheinfrank showed design that was supposed to be original in concept. The design he showed is influenced by concepts of language. There is no real theory behind his approach. What is there is a tenuous effort to pick from various sources in order to produce a coherent design concept. Rheinfrank talked about design language. I would not waste too much
time if I did not see here a danger in the very loose use of the concept. If design language is seen from the perspective of linguistics, then we can expect very little, since design, as well as so many other visual activities, cannot be reduced to language. Over ten years ago, I introduced the concept of design language, for which I claim a copyright. This concept was not linguistically based. I proved that the visual is not sequential, but configurational in nature. Accordingly, I defined the characteristics of a design language in relation to design activity. The issue is not one of copyright, but of appropriate concepts, and this brings us back to my remark on design fascism. If a designer imposes upon his client a so-called design language, i.e., a system of formal relations which the client has to respect in order to get the product to accomplish what the user wants from a product, we have a fascist attitude. It was in this respect that I looked at the examples John Rheinfrank gave of the design work of RichardsonSmith and noticed how instead of involving the user in the design, the user is made a passive subordinate of design; the user is oppressed by the design rules. Fascism, as political system, applied this attitude as well.

Roads built in fascist Italy by Mussolini were designed according to this notion of design. They looked good, but never performed up to expectations. Neither does some of the design of other deceptive systems. And I would be very unhappy to find out that we are on the way to justifying fascism in design. Let me explain this by example. We recently installed a Xerox 6085 machine in our office. And guess what? The design is exactly what John Rheinfrank showed us- neat arrows, color coding, graphic interface, laser printer... lots of cosmetics! Because the system is actually a mediocre product, with low productivity, low level of integration, a system which ignores everything we learned in the last five years concerning user interface. Even Xerox research work on learning is ignored. But the machine looks good, has a big screen. Functions you can get on machines costing half or one-
fourth the price are not provided. The machine looks as though somebody designed it, but in fact it is not a designed machine. It is a "cosmetized" machine. And this notion of design that makes things look good, but does not care about how good they are when used, is the fascist notion I referred to in my remark. This is the Mussolinian notion of design, a system which built highways unable to support traffic. They fell apart once people tried to use them. In the case of the Xerox system, the user falls apart, forced to accept counterproductive conventions and operations so anti-intuitive that one wonders how much of the user profile was considered by the so-called designers.

In reality, the fascist attitude in design goes far beyond the example I just gave. Not only computer technology introduced rules of a despotic nature which force the user to learn how to "extract" some function from a product. Look at the dictatorship exercised by oven manufacturers who produce designs of microwave ovens so complex in their commands that the regular user gets scared, of VCR's, of telephones, etc. I know that the demagoguery of political life interferes with design and contaminates it. However, I would hope that designers will not accept becoming part of the delusion more than they already are.

Linda Ettinger: Mihai, you can't leave with that statement. You have to take two more minutes and talk to people about the ways in which that condition we all accept as pervasive can be ameliorated.

Nadin: Probably my position can be seen as utopian. In short, I believe that design cannot be professed without understanding the nature of the relation between the object designed and those who will use it. In order to account for the characteristics of those for whom we design, we have to know more about how people interact with tools, machines, utensils, and other artifacts. We have to understand that design is not a language but an attempt to structure the environment in ways which
allow for as much participation of the user as possible. Designers participate in the change of the world. This is a very ambitious profession and one of high responsibility. Slogans such as "problem solving" only turn our attention from the nature of design activity. In designing, designers bring to expression structural components of the human being, structural aspects of human interaction in society, and a sense of culture. Part of the sense of culture is represented by the aesthetic component of design. The aesthetic component cannot be reduced to how pleasing objects are, but to how they stimulate creative use and creative interaction. It is not a matter of shapes or colors. It is an issue of sensitivity. Those in the business of making a fast buck imitate the aesthetics of others. Typically, their designs don't last longer than a certain fashion. Good design is design that expands beyond appearances, goes into the engineering implications, and into the cultural function of design: education through use. Design is, or should be, interdisciplinary. It is almost impossible to design from scratch. We receive from our environment so many elements that in the end it becomes impossible to design by ignoring who we are and what we do. Accordingly, we'd better integrate in the work everything that defines us as designers and as human beings. The designer's involvement goes deeper than the "skin" of the product. The new tools we are developing are able to support new ways of thinking. We should pay attention to the way we educate future designers so that they can use the tools creatively. Designers should understand what consistency is, not in order to impose dominating rules of consistency that make no sense, but to be able to search for various ways of maintaining consistency. Consistency cannot be taken independently of the product's integrity. We need not only a concept of design integrity, but also some conceptual and practical tools to implement it. With a Macintosh and laser printer, one can produce all the annual reports in the world. Does this mean that high productivity of the tool is all we need? Before using the tool, we have to develop
appropriate design concepts. This will ensure not only good appearance, but also high quality design.

Rheinfrank: The product that Mihai mentioned, the Xerox 6085 was really where ideas for the Macintosh originated several years ago, the basic ideas. Now, as it turned out, the 6085 was a big failure; Xerox had generated a poorly functioning product, and then they had a designer go out and style it. The design of this machine was done by a different design firm on the West Coast; so, I can take neither credit nor blame for that machine's design.

Ron MacNeil: There is a real tyranny that design imposes on the users of a system; they are often, by and large, incompatible.
PART IV

The Cognitive Style of Design

Discussion led by Ron MacNeil
MacNeil: I’ve been working as part of the Media Lab which is comprised of parallel groups, spatial imaging, epistemology and learning, film, videotape, vision research, speech research, computer graphics and animation, electronic publishing, advanced television research, music and cognition--a lot of interesting enterprises. We have been involved in many interdisciplinary activities, and we are trying very hard to make all of these activities into a discipline with boundaries. It is not clear whether that is necessarily a trivial thing to do.

So, I started my career as a Physics student at RPI, doing photography, and later, I began teaching at MIT and doing experimental print-making which sort of evolved into computer imaging as a way of making the print-making process more malleable. I wound up building a painting machine which was 25 feet high and 75 feet wide and did images the way one would imagine the ancient Egyptians doing them if they had invented video. If you were an ancient Egyptian and had to invent video, what would you have done? Well, you would probably have taken some containers of colored sand, put some sort of stoppers on them, put them on a pendulum and done a pretty good job of making a video. Well, that’s sort of what the gadget does; it is a very large X-Y plotter with spritz heads.

I’ve just invented an optical lathe; I put a microscope on it so that I could work on the small details that one needs for plotting. As it turns out, the plotter is a useful artifact for the world to have, but its most important aspect is its imaging system which serves as the front end of the printer and has a very large virtual space. The net result of this endeavor is that I built a lot of tools to explore the visual environment, but the questions remain: What is design knowledge? How do you capture it? Once captured, how do you represent it in the computer? And finally, having figured out a way to represent it, what do you do with it?
Well, I am not going to be able to answer any of those questions in the first pass, but in an attempt to move closer to some of the answers I will discuss a series of cover designs done by one of our sponsors, Joan Musgrave. Joan is Corporate Designer for Thornwood, a technical design center in New York. With her aid, we managed to get a grant intended to help us find out what an intelligent illustration station would be like. We began by having her bring in some of her cover design artwork. We looked at the work to try to figure out what was happening inside of it, so to speak. Then we asked ourselves the question- “Could we build a system that does some designing on its own?” Begging the question of whether it was a good thing to do, we instead asked ourselves- “Could we begin to do it?” The answer was of course no, but what we built is at least mildly interesting.

Without using any really complicated rules, we devised a little program called GRID; it was sort of our AI environment project. The program started by picking a set of pictures and then arranging them in a sort of random fashion, keeping track of what it had done while applying the four rules that could be identified in Joan Musgrave’s work. It is very simple; the computer uses the images as its search space and arranges them in a number of variations using the constraints that the designer would generate if he or she did this stuff by hand. I think it is probably a very interesting thing to do, and it may be a beginning step in the process of getting computers to design. Right now, I know we don’t know how to do that, but this may be a first step.

Ettinger: Does the user identify the images to be used in the cover design or does the computer provide them?

MacNeil: The user picks the specific images that are going to be used.
Ettinger: Are the images picked from a predetermined set, or did you determine the original set to store within the computer?

MacNeil: Well, actually the designer has to choose the images to be used in concordance with the publication policy. What is interesting is that the examples generated by the computer do not look much different from the cover designs produced by the designer. What I haven’t done is a completely random generation to see if there is any difference at all between the three.

We have about four graduate students now and about 30 undergraduates working on various research projects. We tried very hard to get graduate students who have either come from a design background and have acquired computer skills recently, or ones who have a very strong computer science background and have somehow come up to snuff as a designer.

Wilcox: Do you have any designers working in a 3-dimensional medium, or is it primarily 2-dimensional, graphic design?

MacNeil: The sculptors don’t tend to find us. They don’t seem to need this way of thinking about things, and they don’t tend to engage in the type of modeling we do.

Wilcox: Do you know why?

MacNeil: Well, first, we are not very good at getting things out, and second, and perhaps more importantly, I do not have a sculpture cutting device. It has been dismantled. Now, in fact, if we kept it going, and made our products truly 3-dimensional, it would be likely that we would have some sculptors. Those people tend to want to build things that they can touch, and right now they cannot do that in our lab.
This is sort of a first pass and, in fact, just pixle based. When you make a change in this sort of environment, the only change you can make is pixle wide change. This approach was very wrong, and, pretty quickly, we began to see the limitations of doing it that way. When, for example, you map the thing into a different shape, the relationships are not right anymore; something very seriously wrong happened to them. So, the approach that one graduate student has taken involves, first of all, breaking the thing down into its parts, defining it as a family, or set of relationships and then building a hierarchical constraint environment for representing it. For example, if you were designing the packaging for a cereal product, you might say that the title of the package is at the top of the hierarchy and the illustration is next, the list of ingredients is on the bottom right side, and so on. Now, I'll pretend that in one sense, and I think in a non-trivial sense, what has been generated using the hierarchical system of constraints is a design, but whether we have captured a design in some reasonable way inside the machine, I don't know. This is possibly a beginning point.

One aspect of the thing designed is its constraint environment, and we add to that a knowledge base which involves how you transform what the marketing strategies are, what packages are, and what visual design is into a product.

It is not clear that for individual design projects we really have a terrific gain by expending the effort of putting a lot of knowledge into the machine. If you only have four things to do and, in fact, they are fun to do, why not just do it by hand; there is no real leverage in a design machine. But if you have 3,000 things to do, like the recent Black and Decker job we had, and the variations look alike, why not put that in the machine? We found that when there were 3,000 products, it was about even-steven, the amount of effort to do the job by hand as compared to the amount of effort to put the knowledge base in the machine. The thing that interests me is the scale of the project required to make the effort worthwhile; I
don't know how many jobs there are that have more than 3,000 pieces. So, maybe we should make knowledge acquisition easier. Maybe that's the lesson in all this.

Sanders: Of course, when a project is repeated, by the second or third generation knowledge acquisition will be easier and a savings in time and effort can be realized.

MacNeil: A fairly interesting project, done by one of the vision researchers that joined our group, was an attempt at diagram understanding. The notion is, that if you develop the right set of visual properties along with an understanding about the ways to combine them, and the ways to traverse the tangle that you create, you make generators and recognizers that can live in some powerful processing machine you may, in fact, have the beginnings of machine vision, or at least machine recognition. The method that this researcher used involved looking at the work done by a Russian psychologist, and used the numerous sets of items he constructed which have one distinguishing feature. The vision researcher working with us has been at this project for about two years, and she is beginning to make some real headway. If the task of recognizing an image involved a single instruction, "Look at an image," one could allocate a pixel processor and say, "Everybody look to your right," "Everybody look to your left," and so on. This begins to become interesting only if there are 100 of these pixel processors that can fit inside your hand and live inside your desk, but already we're talking about a million dollars; that's not terribly useful. This again points to what has become an article of faith for me - that art making begins with the tool-making process, and what I've been talking about is the tool-making process, the ability to digitize an image which once digitized can live in our very own computer system.

The first article of faith is that to start the art-making process, one begins by making tools. The second article of faith is that designing is programming.
can do it outside of the machine, but you want some leverage from the machine, then somehow you’ve got to get what your doing inside, or you can take a slightly different approach and begin by doing it inside the machine. Why would someone want to do it inside the machine? Well, I don’t have any concrete answers, but my feeling is that it is probably a profitable thing to do. There is not a lot of support for this position yet, but I believe it will be there. I am going to propose that we will need a new architecture for design machines, and the first thing that you should say to me is, “Why do you need a new special architecture?” “Why don’t you just enumerate all of the parameters that you would ever want to possibly tweek in the search space? Lay them all out in a menu, and, when you want to do this new function, just traverse the tree and set all the parameters.” I think that a 30 meter long menu to do this sort of thing is probably an understatement.

Given the present state of affairs, in most cases somebody else designs the parameter set for me to use. This amounts to someone else telling me how to think about the stuff that I am going to do. Well, I’m sorry, that is wrong. So, what I am proposing is a sort of 3-tiered architecture with the goal of building and manipulating intelligent objects. At the top level, there is a direct manipulation paradigm. Typically, we point to the objects that we want to manipulate and manipulate them directly. They have handles, so to speak, and we can do more or less what we want, by means of direct manipulation. That is the architectural level that we know pretty well. At the next level there are tools and boxes to put the tools in, command boxes, graphics boxes, and so on; all pretty straight forward. At this level, there may, for example, be a brush editor that knows how to make brushes. We have laid out the parameters that we need in order to make brushes, and we can make transparent ones or dynamic ones. At the bottom level, we would want to be able to open up the part that needs to be edited. Unfortunately at this point, we are probably talking about something that closely resembles
programming. The designer using the system would have to think algorithmically, but at least they are protected from the things that are nasty about programming — the things that keep him or her from thinking about the stuff which is really at issue. At a certain point, one would simply not be able to conceive of or understand what is going on at other levels; that is the problem with conventional programming. Hopefully, what we will do is build little modules, contain them and name them and keep on layering, so at some point, the designer will have a reasonable statement which is recognizable in terms of what he or she wants to do. If they want to edit it, to open it up again, the direct manipulation paradigm is maintained, and they have been kept away from all the nastiness. What I have been describing feels like it has to be close to true, but I haven’t proved it yet. I should be able to in the next couple of months.

Whitney: Have you tried to operationalize it yet?

MacNeil: I have got two students who are building this machine for me now, and at the end of this year, I should have some idea of what it will be like, but then again putting it all together is the scary part. I believe that it in fact does want to fit together; it truly wants to be layered. It is really a pretty complex program and that makes it difficult to see on all levels. I think it does have some problems that I don’t quite know how to solve, but this is an initial experiment; it gets us a syntax.

The last issue I want to examine is the question of whether multimodality inputs are a useful thing. We point, touch with our fingers and do a lot of typing. My intuition is that if you screamed at the thing, while pointing, or said “bigger” or “no” or any other command you could get more work done. It just sort of feels right, and it is not that hard to do. Eye tracking is another potentially useful input source. One thing that kills me is that when I look at a particular window on the screen, I should be able to type into it directly, but instead I have to move the
mouse in order to type in a particular location. I have already looked at it, so why
doesn’t it know that that is the window I want to type in? Scratch pad, or gesture
recognition input seems to be another useful thing. I certainly know that I want to
edit text, and I would like to be able to go in with a big, thick pencil, and I want to
say "no" right on top of this very careful formatted text. Multimodel input
certainly seems like a good thing.

Gorski: At the outset, you discussed how you developed hierarchy of items. I have
noticed that in your pattern discrimination work, size is used to represent status.
Aren’t there other pattern relationships?

MacNeil: Absolutely, but this was the most dramatic example that we could find.

Gorski: We learned that size represents status, but there are certainly other ways
to represent it. You can create status using overlapping relationships. Do you have
anyway to enter alternative systems of categorization into your machine? For
example, the concept of overlapping relationships as a way to express status or
express importance. Can you implement this on your machine?

MacNeil: The answer has to always be yes; that is the only thing I can say. But it
does not always come easily, and that is a very important point. What I would like
to do and see happen pretty soon, is to have some designers come in and make
changes like that and say, “No, you’ve done that completely wrong; I want it to be
like this.”

Gorski: Your status is organized on the basis of the size of the figure and that
determines its relative importance, but what about the size of the ground? We
utilize the ground as an attribute all the time, even if we are unable to give it an
adequate description.
MacNeil: That is a crucial point, and very subtle. I think it has to do with the fact that we concentrate so much on the thing. We spent so much effort making objects that it's easy to forget that that is only half of the problem.

John Emery: One statement that you made that I find disturbing is that you believe that art starts with tool making.

MacNeil: That is the way it happens with me; my art starts with tool making. I think that it is a style of thinking. When you build something, it starts as an idea; it has a particular form, and you assembled some tools to make the thing. If you are a video artist, you assembled the tools to do it; if you are a painter, you assembled a different set of tools.

Emery: But the thought process in art must come before the tools.

MacNeil: You are not going to have an idea which is completely outside of the way it is going to be made? That is part of the concept—the environment that the thing gets created in. So, there are a set of tools that you assemble. You make a particular piece, and you make variations on that piece with the tools that you have. At some point, those tools are no longer right. They outlive their usefulness; they have to be transformed.

John Davies: Since I'm an ex-designer, I'll carry this example as far as I can with reference to film making. Say you take a copy of my film; now it's one thing to simply repeat the same thing, but there are a lot of variations on what I've done. What if I packed up the videotape and came to see you and said, "Here is my product; what can you do to help me see the variations without me physically repeating the whole process?" There is nothing that I know of that resembles this type of film that has been done before. But, in my case, I did not invent the tool, I
went and found someone who knew how to use their tools with my ideas. Now, if I don't want to go through doing that again, but I would like to make ten other variations of that film, how could you assist me?

MacNeil: What you want is the leverage of the conceptual tools that you built over time. They want to be there as part of your work, and I'm beginning to explore what the conceptual tools might look like for design in general.

Emery: But you are also limiting your art by beginning with a fixed set of rules that are applied repeatedly.

MacNeil: I certainly am, but it is a start.

The focus of my research, by and large, has been to make the computer habitable for designers, and in particular, to determine what a design workstation would be like. What I have observed is that when designers build their design, they build the whole thing, and then they give it to the production people. The production people analyze it, segment it, split it apart, divide it up into lots and lots of pieces and expend a huge amount of effort doing that while adding no new information in the process. Everything that the production people do is a rehashing action. Rehashing and repetition is what computation is all about, so it seems possible that one could bring the design process into the computational domain and create computational designs.

Ettinger: I think that the layering effect that you were discussing is very exciting because it allows multiple entry points for different levels of users and that intrigues me. On the other hand, when you define information that will function in the system, and then organize it in an hierarchical order and structure it in terms of a matrix, you force discrete choices. What that in effect does is create a design system that is based on a single world view, and a single way of organizing
information and structuring ideas. If however, the designer chooses to design according to some other organizational system and presents it or represents it in a way that shows interaction among categories for example, or that describes the way in which certain factors fall in more than one matrix cell, he or she is likely to find that it cannot be done within the framework you have described. What you seems to be doing is eliminating the ability of the user to work with multiple meanings, or multiple ways of organizing information, or structuring the world.

I think it is interesting that you say that the system want to fit together and what I'm claiming is that it wants to fit together because we exist in a society and a culture which is very heavily oriented towards a hierarchical system of organization. It is so persuasive we do not even notice it; so, of course, it feels right. If you take it outside of a particular cultural bias, it might not feel so right. In addition, when you say the layering of information protects the user from those messy things that they do not want to become involved in, the corollary to this protection is that it also strips the user of power to make choices and to be responsible for the choices. I do not know how to build the machine that you have been describing and, I do not know enough about information structures to suggest an alternative way of doing it, but I do recognize the problem. Technology it seems, has already given us the capability to explore a variety of approaches, your system seems to offer the user only a single approach.

MacNeil: This example represents only one paradigm, there are many other AI technologies that we have not even touched . I think that the next level in the advancement in the area will involve combining those technologies.

Ettinger: The situation becomes dangerous when a system that is based on this hierarchical matrix orientation of discreet categories becomes thought of as the way it is done, or as the only possible solution. The idea then emerges that this is
how computers work, and it may be so pervasive that people do not know that anything different is possible. It seems to me that as designers, we are almost obligated to present alternative approaches with the technology.

MacNeil: What I presented this morning was my first experiment. No doubt, other very important paradigms need to be tried and when you hear about them, please let me know.

Ettinger: Well, it is not just that they are important to try, but by making the choices we have made to explore only certain directions, we limit the audience, we create haves and have nots, we empower and disempower.

MacNeil: Returning to the point you made earlier, the reason I am proposing this system the way I have is because I have assumed most designers do not want to become programmers. It is not true for everybody, but the kinds of people gathered at this symposium, I suggest, do not want to to become programmers. I work on both sides of the fence, and when I am thinking in terms of images, if I have to stop and think about writing that in a programmable language it gets in the way. It is not the same way of thinking, and it begins to interfere with what I am trying to do with those images.

Ettinger: Does the power to manipulate have to be based on a particular kind of programming? Do I have to give up control and responsibility in order to gain some sort of computational leverage?

MacNeil: What I am hoping is that the user would feel that he or she has 90% of the control available with initial programming languages, and be protected from 90% of the badness. That is the balance that I am trying to strike.
Ettinger: But you were defining my world of choice, very, very tightly with that structure.

MacNeil: No, I am defining my world choice.

Ettinger: You are defining your world, which then limits mine, because I am working within your system.

MacNeil: Well, that is when we want to go to the next layer. I just do not know another way out of it. So, it seem that there are some experiments that need to be done that address the issue of cognitive style.

H. van Dijk: I have a very pragmatic question coming from the perspective of a Mom and Pop store for graphic design. How many programs do I need?

MacNeil: I think that you only need about four or five.

H. van Dijk: Well, I have a program for type, and a program for drawing, and one for paste-up, what more do I need?

MacNeil: One point that I did not make very well, was that the middle layers, clearly need to be structured within the domain. That is a pretty difficult task and someone is really going to have to spend some time thinking long and hard, trying to figure out what that domain is all about. I think that this is only the first pass. My answer to you regarding how many programs you would need, is that once you get enthralled and understand how powerful the program is, you will use for awhile and all of a sudden you will realize that you want to do something different and that the program does not do it. So, what we might like to have is a system with enough flexibility to change as the designer changes in order to stop this endless need to always have one more piece of software.
William Mitchell: There seems to be, underlying this discussion, a kind of desire for there to be some sort of universal design system that anybody can use and that is broad enough for anybody’s needs. Now, one can compare that to the situation that is found in music. One of the great richnesses of music is that there is an immense idiosyncratic diversity of musical instruments and a lot of the richness of musical composition comes out of the engagement of people with difficult, recalcitrant instruments. The instruments are not easy, they are hard to manage and take a long time to learn. However, something comes out in the engagement of the imagination with an idiosyncratic structure. So, in this sense, it would be interesting to think of a world of computer graphics and computer aided design that in fact encourages that kind of idiosyncratic diversity; challenging instruments rather than friendly systems.

Novak: There is an in-between ground which is, to take the example of the music and the instruments, the synthesizer, where one has an instrument, but the instrument is reconfigurable. I think where these two ideas come together is where the user has operators at his or her disposal that can, if so desired be put under the hood. When the person decides to, he or she can get under the hood, change the plugs, put a V8 into the Volkswagen and fly. The “how” is what I am trying to get at. The “how” is a problem and it is not a technological problem.

If I take a bunch of designers, good solid designers, and ask them to design for me a slide projector, a pen, or a teacup, something simple, and require them to do it without their hands, to do it by telling someone else how to do it, what would those designers say? How would the designer express to the other person what it is that they want done? Now, bring in some AI technology, put it in front of the designer, give that designer all of the graphics and all the computational power that he or she
wants, make it run in parallel and give it the capability of following the designer's every command, what does the designer tell it to do?

Ellery: I think that is a very important point. As we address this system that is so malleable and so deep in its richness of knowledge and information, it becomes apparent that we bring to it a language construct or a mind set. How do we work with that mind set so that we can go down different avenues, and explore different alternatives?

Mitchell: Let us assume that we have a machine that is configurable in anyway we want. It seems to me, that what the serious designer really wants to do is rigorously exclude a great many things that he or she just does not want in their world, and that the important thing is to structure an interesting world to explore. So, flexibility, and reconfigurability are enabling mechanisms to let the designer clearly define his or her world. However, it leaves unexplained the question, “what kind of world does the designer want to define?” That is the big problem.

Nadin: This brings us to a point of recognition. You know, and everybody knows that when you go to the computer, what you need is a computational function. If you can not provide a computational function, there is no computer that can ever do anything to it. So the question is, how much of design and what part of design can be represented by computational functions? Is all that design is, capable of being represented by computational functions and if not, which of the segments of design can be represented by computational functions? I leave it as a question, because I do not know the answer.

Krippendorf: I think if we look at history of computation in general, we will see that the computer was invented first, and then people got very fascinated and later we were told that this may be the way the mind works, and finally we began
asking, "how can we use the computer for our purposes?" It is correct; unless we have a computational function we cannot use the computer for the things we want. I think what is needed first of all, is a good design theory or perhaps a theory of how designers perceive. John Rheinfrank talked earlier about the design language, and that is the direction that I think we have to travel. First, understand and develop the language of design, so we can talk about design, about procedures of design, and then see which part of that could be described in terms of an algorithm.

Let me return for a moment to this morning's exercise of designing an expert system to aid designers. Unfortunately our group did not proceed as predicted and we could not therefore, present what we had discovered. I think we discovered quite a lot, so I will take some time now to describe what it was we discovered. It pertains to what goes on when one does design. We started by trying to come up with a design problem and after some discussion we somehow decided to design a new bed pan. There are already well known concepts for this item, which if used, would have resulted in the design of another bed pan. We decided that this is not what we wanted to do, we did not simply want to retrace, or make variation on an old design. Instead, we began by looking at what the bed pan does. This means, we completely destroyed the concept and then reconstructed a concept focusing on what the things does and what function that it serves. This is perhaps a laughable example, but it shows some of the processes involved in design. We started out with a very general and perhaps vague concept, and then listed all the knowledge we had, from physiology to immunology, from chemistry to physics, which would be required to design the bed pan. Then we wanted to contract the problem because it had become too large and then expanded again, contract it and expand it, over and over again. This process of listing knowledge that would be required, finding it to be much, then narrowing it down, then expanding it and then narrowing down again, expanding it further, and then narrowing it down again is the type of process
that we typically engage in when we do design. I, therefore do not think that a
good way to approach the problem of computer aided design is to begin with a
computer and say: "here is a computer, how can it be made useful in design?"
Rather, we should begin with the question concerning how we design, and then
ask, "how can we describe it in such a way that we agree on what is going on?"
After we have somehow come to a consensus, then we can ask which parts can be
written in terms of computational function. We might come to the conclusion that
only a small section of our current knowledge can indeed be encoded in a
computational format and that the rest ought to be left to human cognitive
processes. I think we should shift our focus to issues concerning how people think.
Thinking should be the center of our attention, not computers.

Ettinger: It might be better to ask, " what is the appropriate role of the computer
in the design process?", rather than, " how do we let the computer control the
design process?"

Guindon: The problem with that statement is that the design process is partially
controlled by the tools that are available. There is nothing that says that we have to
think in a certain way from now until forever, but the tools available will certainly
shape the thinking process.

Davies: It is apparent to me that there is an enormous gulf between what Klaus
Krippendorf has just described and where you (Ron MacNeil) are at. I know in my
own case, when I was designing, I would often take an antitheoretical point of view.
I believe a lot of designers think that way. In other words, whatever the rules are,
we want to break them. Whenever someone says, "you can't do that," we want to
do it. I think this is a very important part of the structure that we are talking about.
So, what I am saying is that there is an enormous gulf between the way our minds
work and what you have been saying about computers doing design. I wonder if there is some way that we can bring these two things together?

MacNeil: The AI community does not have any problem with what has been said. They have theories of perception and cognition that can be modeled. It does not really seem to be a major problem. My gut feeling is that an awful lot of things are ultimately computable; one just has to be clever about doing it.

Krippendorf: You are making it a quantitative argument, when really I think the question is what is that that can be computed? This is what we have to start working on.

MacNeil: The start of that is some cognitive theory and within that, or touching that is some theory of design.

Nadin: Dr. Chandrasarkaran has suggested that there is one class of problem which can be described, more or less in the way Klaus Krippendorf suggested that we look at them. The first class of design problem, according to Dr. Chandrasarkaran is the truly creative design class that cannot be dealt with from a computational perspective. The second class of more or less creative design problems can also not be dealt with from a computational perspective. The only thing computers can currently handle are routine design problem.

Novak: As a way of approaching a theory of design, I would like to suggest that any kind of theory of design ought not to be exclusive, but rather has to be inclusive. You find your bag of tricks and you keep putting things into it, and when you come up with a new one you say thank you very much and put that one in too.

MacNeil: Hopefully, that bag will hold a lot of tricks.
Mitchell: I think it is possible to claim that what one wants is the smallest possible bag of tricks to be a good designer. A good designer needs to know exactly what he or she wants, and refine the bag of tricks, right down to precisely what is needed, and absolutely nothing more or less. A similar approach can be taken in building computers that will aid designers. I think there are undoubtedly some very different styles of design which may best be described by different theories. Classical architecture, for instance, is a kind of architecture that thrives on the existence of a very precisely defined system of rules and a bag of tricks or operations to work with the rules: Design is a refinement within that system. There are, however, other styles of design that depend on an array of complexities and nuance, a very complex cultural situation and the blending all of these elements together in subtle ways. I am making an argument for the respect of diversity of styles of design, and this meaning is going beyond simply reconfiguring a computer system.

Dilnot: One example of design that seems to be an ideal form to put on a computer is classic architecture. Classical architecture is a system of carefully formulated rules that could be simply applied to a computational framework and therefore, one would think reasonably enough, that it would be possible to design perfect neoclassic buildings. However, I very much doubt whether one could do that because design often is a subtle interplay between a set of rules and reality, regardless of whether it is an image or 3-dimensional material reality.

MacNeil: How much sensory input do you need to feel in order that you really feel that you are in a place? If I give you an exceptionally clear stereo view and a tactile feedback so you can reach in and touch the thing in front of you, at some point, you will believe that you’re touching the real thing.
Dilnot: This is where I say no, absolutely no, because you are no longer talking about posing an object or a building against recalcitrant complex reality, but instead you are talking about creating a defined, simulated reality, a reality that has to be pre-defined and pre-determined and therefore, astonishingly limited compared to the real thing in the real world.

MacNeil: Klaus claims that the only important things are the perceptions that occur inside his head and the light that reaches his eye.

Dilnot: I see the computer becoming not so much a tool, but something that is beginning to determine the design process, and as a result we are not going to have an increased amount of design knowledge, but a falling amount of design knowledge. I think one can see this tendency for a decrease in the amount of design knowledge already in existence.

Todd Cavalier: I think that is a point well taken, and there is something that I would like to add. I saw a piece of machinery the other day that does spectacular animation. The things that it does are so engaging, that it is very difficult not to be carried away with the equipment. I am wondering, because of the ease with which the user can approach this thing and create all sorts of spectacular effects, at what point are the fundamental skills and fundamental knowledge of designing with the formal relationships between the various components that one is working with, lost, abstracted, or made artificial? At what point should students at an educational institution be allowed to handle this incredibly engaging machinery? Should they first have to develop the hand skills or should we just say, "forget that stuff, it's not necessary anymore?"
Krippendorf: I think we are confusing the issues. The first is, what role should the computer play? I think if there is a very engaging computer program, why not let the students play with it? The other issue is, what is design and whether knowing what it is helps? They are two separate questions and I think we should not confuse them. I would prefer that we not talk about whether the computer is good or bad, but rather about what designers can do with it and whether it is a constraining or enabling technology. This is a very crucial question. I think that if one takes the computers very seriously, they can be very constraining. Because one has to do so much programming and because it takes so much time to acquire the programming skills required, that even if in the end one is disappointed with the results, he or she is likely to feel that so much time has been invested that the things must be made to work. The result is that the designer is stuck with a very limiting technology. On the other hand, maybe the problem can be approached differently. Instead of letting computer programmers define the design domain, let us as designers describe what we really do and what the computer programs do, and then see which activities can perhaps be programmed.

Nadin: You are right in making this point. However, one of the reasons we discuss the computer at a symposium about the cognitive condition of design is, that the computer may be one of the first attempts in the history of design to build some of the conceptual tools necessary to understand design. Computers besides being very useful tools, happen to have very interesting epistemological realities. By simulating various phenomena on computers, we are able to gain knowledge in areas where further advances at one time seemed unlikely. I think that in trying to find out how a designer uses a computer, we are at the same time lead to question, “how much we actually know about design in the first place?” and “how much of that knowledge can or cannot be represented?” I consider the computer, at the
same time, to be both, a conceptual tool that is a reflection of the designer, giving him awareness of what he is doing, of what he wants to do and of his own limitations.

MacNeil: We already know a lot about design.

Nadin: You might know more that I do.

MacNeil: We all know a lot about design, what we do not know right now, is how to get that knowledge into the machine. We do not know how to do that.

Nadin: I claim that we do not know a lot about design, and that is one of the reasons why I said, we need better theories of design. I do not believe we know all that much about the cognitive condition of design.

Gorski: It seems that most current theories of design carry with them an assumption that we all understand the experience of human sight. Yet, as I listen to the different descriptions, I have not found a commonality that would give me the assurance that we understand the nature of visual perception. We ought to be aware of the fact that every human being's experience of sight is likely to be similar to our own. The range of meaning might have cause to vary more widely, but we still have not completely understood the common visual experience that gives rise to that meaning. I would also like to point out that very few computer programs have gone much further than the vernacular concepts or uneducated assumptions that are held by the layman on the street regarding the nature of visual experience, and I do not believe, that we as a profession, have a good handle on how meaning is derived from the array of visual information. We might, in an effort to reduce this deficit in understanding, look to the authorities in experimental psychology; they are still coming up with the answers.
Wilcox: I would like to comment on something slightly different. It seems to me that we are in the middle of a process with the computer. It is a process that we have gone through with many technological developments. I would not be surprised if fifty years ago, there was a conference on the effects of the electromagnetic motor on design. At that time, when the new technology emerged, the presence of the technology in the design was very apparent. People were undoubtedly asking themselves, “how can we use this motor?”

There was something that people use to talk about called computer literacy, something which people are not talking about very much any more. I remember not too long ago when everyone was saying, “this is the computer age, and if you’re going to get anywhere in life, you will have to know how to use these things.” So, many of us sat down and learned TPL logic, and how to do analog to digital conversions, and within a few years, all that knowledge was obsolete. What it meant to be computer literate, had completely changed. What it means is still changing very rapidly and I would suggest, that within a few more years the very concept of using a computer will have disappeared because the computer itself will be hidden within the tools that we use, in the same way that electric motors are now hidden. In the same way that we do not have to be a mechanic any longer to drive a car, we are soon not going to need to have any kind of special knowledge to use computers.

MacNeil: That may be true, but I do not think that it is going to help us much with the problems we are wrestling with right now. The fact is that we know a lot more about design that we are able to transmit.

Brian Zaff: We, in essence, need to be able to formalize some aspects of the design knowledge.
MacNeil: All the people that have learned design went through a process of formalizing, internalizing and questioning their knowledge over a long period of time. During that process, one might in fact have a chance to look at the process as it is unfolding.

Krippendorf: There are different kinds of knowledge, there is knowledge about the world, and there is procedural knowledge. The latter type of knowledge, procedural knowledge, is not readily verbalized. For example, I have the ability to speak English, but I cannot specify, in a precise manner, its grammar. I have embodied knowledge, but not to a knowledge that I can pass on. I think that much of our knowledge of design, in many respects, is very similar. We can do it, we have acquired some knowledge, but we have not learned to formalize it, or how to write it down. It seems to me, that the process of formalizing design knowledge is so far away that we do not even have the concepts available that would describe how we should proceed. The efforts being undertaken at RichardsonSmith with the formulation of a design language, seems to be a step in the right direction. It is an attempt to develop a language, a set of terms, so that there can at least be a dialogue among designers about how to design things. That is very far removed from a formalized design being put on the computer.

Zaff: Short of formalizing the entire design process, I believe, that it is at least possible to identify one element of the design process which could be formalized, and could therefore be implemented in the computational form. If we sit back and try to think about why designers design, and what are they trying to do when they design, I think we might get some clues as to a particular aspect of design that can be formalized. What I am hinting at is, that people design in order to change the environment, to make things that benefit them more present and things that cause
them harm less pressing. This does not include the aesthetic element of design, but it does include the functional element, and that, I think, seems to be a very important element in every design process.

Krippendorf: Could you write an algorithm?

Zaff: Certainly not yet, but I have been working during the past month on using the Gibsonian notion of affordances to characterize the functional aspect of design. The concept of affordances makes claims concerning the presence of lawful relationships between the object in the environment and the person using that object. If there are indeed, as the concept claims, lawful relationships between the person and his or her environment, and designers are in the business of manipulating these relationships, then I would conjecture, that because of the presence of these lawful relationships, it should, at least in principle, be possible to formalize these relationships and thereby formalize at least one aspect of the design process. In practice, the process of formalizing this relational knowledge will be quite difficult, because by definition, the knowledge will be specific not only to the design domain, but also to the particular design project within that domain.

Guidon: There are some cases that demonstrate the possibility of formalizing the process and the domain specific information that is necessary to a particular project. Admittedly the programs are not very good yet, but they at least demonstrate that such relationships can be captured in a computational format.

Krippendorf: There is a major problem with trying to get computers to do what human beings are capable of doing that arises because of the crucial difference between the way the computer works and the way the mind works. The idea that artificial intelligence has as its underlying metaphor is the human condition. The first computers were thought to be emulating the behavior of human beings. What
has become apparent, is that computing is only one of the things that the mind does. I am happy to say that the more we compute, the more we see that is not the way a human being thinks.

Zaff: I would be the first to agree with you on that point and it is clear, that the longer we work with computers the more we realize that the human brain is not synonymous with a computational device. I was not suggesting, however, that formalizing one aspect of the design process will enable computers to begin to do what the human designer can do. The point I am making is that, to the extent that designers are manipulating lawful relationships that exist between the person and his or her environment by changing the environment, at least one aspect of design can be formalized. The computer can be given knowledge of these lawful relationships in the form of a data base, and the computer can then facilitate the design process by keeping the designer informed about this formalized aspect of the design process, as the designer goes about solving the design problem.

Dilnot: There are a few remarks I would like to make regarding what I see as the direction of design in the world and what designers think of it. There are a number of ways of looking at it, but I want to focus on one thing that concerns me in this, so called, post industrial world. It seems to me that we are living in a different kind of world that has changed on a number of dimensions in the last forty years. The project for design gets larger, and it gets larger in a historical sense. There was a time in human society when things tended to repeat themselves. Cultural traditions acted as a stabilizing force resulting in a period of very slow change which, by in large, lasted until the end of the 18th century. I think we then go to a period that we can roughly call the modern period, where there is an increased tendency to reject the past and replace it with an implicit notion regarding the future. Particularly, in the 20th century this implicit notion has taken the form of an
assumed technological determinism, which will follow a single course. If there is anything that postmodernism, or more precisely postmodern thought has brought about, it is a sense that we have not only, to some extent, liquidated our past and are doing so everyday, in all sorts of ways, but that we have also liquidated the future as a simple direct projection forward in time. The future no longer exists as a single one-dimensional entity, and it is no longer possible to make a projection of what America will be like in fifteen years time. Thirty or forty years ago there was a sense of an orderly progression in our development. Now, we no longer have that. We have a continuing process of modernization, indeed the hypermodernization that is occurring on a world scale, and to some extent we have become free of constraints. We have lost the constraints of the past, and we have lost what the past enabled. We have lost the notion of progress and lost what that notion enabled. We find ourself in a new situation where theoretically, at least, our society is more and more the product of our own invention. In a sense, our culture no longer depends on our past, there are no pasts. We have liquidated so much of our tradition, and the replication of our past, that to some extent, we are faced with the task of inventing the future, our future. This means that we have to find consensus ways of determining the future. The future is not at all predicated, it has become a matter of choice, and therefore at a philosophical level that is wider than we have so far discussed, design becomes extremely significant. In a sense, what else does design now become, but the definition and actualization of what it is that we would like. I am not talking about a utopia, as they generally imply a brake with history, but rather in a much more pragmatic sense. There is this sense in which design as a mode of thought, is forming politics and decision making. One may argue that design is becoming extremely important, important in the way that technology was important a century ago.
Novak: Are you saying that we can design the future? If I heard you correctly, let me point out that there is an inherent paradox involved, because as soon as you say, that we design the future, you are saying we design multiple futures. If there was only one future, then obviously we would not be designing it.

Dilnot: I think you are quite correct in pointing out that designing our future implies that we are making multiple futures possible. I think if we do not accept that condition, then what we get is simply the continuation of what we have. A few years ago, I think we could have convinced ourselves that what we were going to get would, by and large, be better, that was the opinion of the 1950's and 60's. Now, we do not believe that, we do know full well, that we are going to have to intervene into the management of technological systems much more than we ever had to before. Otherwise, our entire survival is in jeopardy. We are going to have to intervene in the management of technology in a way that we have never previously done and with the conceptual tools that we, at present, seem to lack. This deficiency results, it seems, from the logic of technology which, at the moment, excludes precisely those concerns from the technological development. Technology's logic, at present, is concerned almost exclusively with the development of the engineering performance capacity of that system, regardless of its potential to cause harm. We have a great many problems and need to find ways of managing the technological development, and that seems to be the future role of design. Design needs to begin to conceptualize the process of bringing the wider human concerns to the development of technological systems. I for one, cannot see any other alternatives to the task of turning technological systems into design systems. As far as I can see, that is what we are going to have to do with large scale technological systems. One has to be able, not only to deal with technological
questions in the inception of the technology, but also with ethical questions as well, and it seems that we are very ill-equipped at the moment, to deal with the latter.

Gorski: Are you suggesting that design has to be turned into management?

Dilnot: Yes, maybe it needs to be.

Gorski: Design requires the concentration and integral support of many different professions, and Clive, you are talking about the need for formalizing our own address to the many social and ethical problems that we all face. Actually, we already have many formal ways of addressing these problems that come from the disciplines in which we are often in a position of authority to say, “these things,” whatever they might be, “will or will not occur.” One of the issues that we all face as designers is the meaning and function that the objects we design will have, because that is where society interfaces with the design. People have to look at an object and understand how to use them, it is our obligation as designers. This is something that we have not really approached as a profession.

Ettinger: I hear two things, one concerns the design process and the other concerns the design. I think that design process needs to have the managerial components within it. What I hear Richard Gorski saying is that the design itself, is something other than management. I think the design process is the more global of the two, and that the design process includes design as well as many other aspects.

Sears: There are many different disciplines attempting to solve some subset of the world’s problems, but the profession that is primarily worried about beauty in the built environment is the design profession. I have heard designers trivialize what they are doing and say, “I’m only dealing with aesthetics.” Well, motivating someone to use a product correctly is not trivial, it is the key to the product’s success.
The motivational factor is as important as the functional or human factors that go into marketing a product.

I have a question, with a slightly different focus that I would like to bring up for comment. Are designers the people who construct the little network webs upon which the subdiscipline specialties hang? Sometimes it seems that way.

Ettinger: Well, designers used to do that. The modernist conception of the world was based on the notion of the purity of different spheres of culture. For example, one could have form and content, or one could have the method on one hand and the meaning on the other, they were considered separate. The postmodern theory has expanded that notion. It has enlarged that notion and ultimately led to a rejection of that idea, in so far as we began to see that if one thing is happening at point A, its effects will be seen at point B. It has become apparent that we no longer exist in a world where we can legitimately, in either practical experience or theoretical debate, separate such things as ecology from package design, or world hunger from marketing issues. We do not have the luxury of a naive world view anymore, it is gone. From that standpoint, the postmodern maybe an end to the modern. The emergence of postmodern thought, however, also opens up a lot of new opportunities which leads into the notion that we are talking about, the ability we have to design our future.

Mihai's discussion earlier about the value of simulation as a learning and investigative tool for discovering what we do when we design is intriguing me. I think we need to try constantly, to strive at reaching a balance in which we find both a computer technology that is shaping human awareness, and a human awareness that is shaping the development of computer technology. Maybe we have been leaning to one side too heavily, maybe we need to let human awareness gain some control over the development process.
Guindon: There has been a lot of mention about the design process as if there is one design process. I can think of at least three distinct processes that I have observed, three very different processes within the same discipline. One process involves design by example, where the designer retrieves several known examples of the product that he or she is designing and modifies them in order to produce a new design. Another type is by transformations where the designer knows a set of legal transformations, has a very precise description of the goal state that he or she is trying to reach, and a precise description of the current state of affairs. In this case, the designer simply applies transformations in order to reach the desired goal state. The third design process is one of prototyping, in which the designer simply builds something in order to see whether it does what he or she wants. Often the designer does not know exactly what he or she is trying to achieve, but instead sees if what they have come up with is something that is acceptable.

To support this claim of multiple design processes I can summarize the results of a study in which I was involved. In this study I looked at the ways software designers came up with programs for the operation of an elevator system, an elevator system which included several elevators serving multiple floors and efficiently handling simultaneous requests. A good solution to the design problem involves the use of a distributive system. One of the designers in the study, did not know much about distributive systems, but he was a good software engineer. His solution made use of general techniques of software design without any specific design schema upon which he could rely. Another designer having knowledge of distributive systems was able to very rapidly recognize, from the description of the problem, that it could be thought of as an instance of a distributive system. He was able to apply his design schemas, and decompose the problem rather systematically into a set of subproblems that could be easily solved. The third designer who we could call a
hacker, just went and did something fast, looked at what he had done and what
we wanted as a solution, and then just kept on debogging the program until the
solution was achieved. So, there does not seem to be just a single design process,
but rather there are many alternatives.

Sanders: I would like to add something that follows directly. Listening to all of this,
it does not seem if we know if there is one design process, or if there are many?
Can it be computational, or can it not? There is however something related to
design that computers are real good at, and that is holding alot of information.
The one thing that seems apparent is that people know what kinds of information
are relevant. The kind of information that is globally relevant, and the kind that is
relevant for a particular task, so maybe we should look to the computer to hold that
information, to store it, to make it accessible to the designer, and leave it to the
designer to use the information in a design process. We know computers can do
that and we do not know about the rest. I am not aware of any data base of
knowledge related to design, but I think such a data base could be compiled and I
think it would have a tremendous utility.

Nasar: I would go a step further than that and say that in their present state,
computers are able to go beyond just storing information. I think it is possible to
formulate a program based on our knowledge of perception, cognition, human
spatial behavior, and the various building materials and manufacturing
technologies that could function so that when presented with a particular design
problem it could inform the designer of the various trade-offs that he or she might
be facing. This might be a way of making explicit the numerous interactions. The
presentation of this information need not be just a verbal message, but could
assume a graphic format. The designer could then work toward the creative
synthesis of those numerous interactions. The present capabilities are there, but
the question that I think this seminar is trying to address is how one goes beyond that, how does one get a computer to do the creative synthesis, or can it be an aid in that process? This seems to be the artificial intelligence question.

H. van Dijk: I do not appreciate the desire to eliminate my daily activities. I wonder whether if at the time of Gutenberg’s invention of the printing press a group of scribes sat around discussing how the printing press was going to change their way of life.

Wilcox: There are really two basic issues here regarding computers which, although related, are quite different. The first is, computer as metaphor or model of what a designer does, or in the extreme, the computer as replacement for designers. The other issue is computer as tool. What tools do we as designers need to help us do our job, what tools are available, and what tools would we like to see become available? My own bias is that the first view is almost wholly pernicious. That is, the computer as model of the human being, is a big mistake. It leads us to ask the wrong kinds of questions; it makes much more sense to simply ask, “what do designers need and how can we give them that?” The two questions overlap to the extent that we might say, “there are things that I do now that I would like the computer to do.” At this point, it might make sense to model certain things that designers do, so the computer can do them instead. Still, I think it’s important to focus on the computer as a tool like any other tool.

Davies: Would it be possible for us to define what we do as designers in two terms, that is, the rational and irrational? Could a designer sit down and say, “here is what I do that is rational and here is what I do that is irrational?” To me, the irrational would be all those things that the designer does that are unpredictable, and therefore, it might be those things that we want to hold on to. The rational
things are the things we need help with and the things computers are good at. Let us go back to our basic question, "what is design?" How much of what we do is irrational? How much of what we do that is new is irrational? As an answer I suggest that almost everything we do that is new is irrational.

Nadin: The word irrational, comes again like the word fascism, so before everyone gets upset remember that the word "rational" comes from the word ratio. Ratio means what I measure, irrational means what I can not yet measure. There are some elements of what we do that are irrational which means we do not yet know how to measure them, how to introduce our ratios.

Sears: We are, as it were, pre-programmed to seek unpredictable things and seem quite clear, that we get reinforcement from surprises. Now, as a designer, it is in our interest to put surprises in the product. Surprises are attention getting, and we bring value to the product by making it gain attention. I do not want to enter into a debate about semantics, but it seems to me, that putting surprises into a product, something unexpected or novel, would have to qualify as being irrational in concordance with John Davis's definition.

(speaker unknown): The frustrating thing for me is that we are constantly being asked to create the unforeseeable, irrational results using strictly rational processes. The only use that I have for a computer, or anything like it, is for replication. Think about what a program is, it is the restatement of something that I already know. It involves breaking down that knowledge into numerous parts and allowing it to be replicated, and this process is not going to get me something different. One cannot create anything novel without building in randomness and randomness is just what we do not want to see in a computer. As soon as it is clear that the designer cannot describe all the parts, and, cannot enumerate all the steps, the computer scientists
loose interest because they need to be able to predict the process. They want to be able to predict an unforeseeable outcome and that is, pure and simple lunacy.

Sears: I agree with you, but computers, nevertheless have an obvious utility for designers. A computer is good for keeping track of all the things that we should not do. The mismatched door that does not fit the frame, the human factors that does not follow a logical sequence, the color that can not be applied with texture, and so on. Our work is so complex that we are bound to forget some things, that is why we have 50,000 notes in a project folder. Using the computer for this type of overhead is nice, it takes the burden off of the designer so the designer can put his or her attention elsewhere without having to be paranoid about what has been forgotten.

O'Grady: It's really part of a larger thesis. I have come to a number of meetings like this, on AI, computers, and design. Everybody sits there talking about the computer as a model of the human designer, and all along, it seems that people are missing the very simple point, computers have not been applied well, even in the generic sense let alone to AI applications. I was consulting for a design firm a short while ago and I brought up a point that their accounting system was in need of some serious review. Although we have not been discussing the practical concerns in running a design form, it is worth mentioning that most design firms go out of business, not because they do not have enough business, and not because they cannot design, but because they do not know how to keep track of money moving in and out of the office. The point is simple, there are a million programs out there for running a business and none of them fit the design business because designers have not written the programs. There are programs for everything one would ever want to do without even considering AI, but most of them do not fit design because designers have written their own programs. In order for us to be able to
make the computer work for us, we have to, once and for all decide, what is
different about what we do? What is different about how we do it? What do we
have to know? What don’t we have to know? Where should we have checks and
balances? So, why don’t we get that which is codifiable and subject to very strict
rules into the machines? Let us get that stuff out of the way, so we do not have to
worry about it every morning.

Novak: I agree with you, but I would say that the fact that the generic systems have
not worked for designers may not mean anything more than that design is the most
complicated application so far encountered, and that computers and their
programs need to evolve to meet the more demanding and complicated
applications.

MacNeil: We have to wrap this session up. You all are invited out to the Media Lab
in Boston to try out these ideas. There has been some very good ideas tossed about
this afternoon and I would like to see something come of them.
PART V

Computer Graphics and the Exploration of Design Ideas

Discussion led by William Mitchell
Mitchell: The field of computer aided design (CAD), and computer graphics as related to the design profession, are not new ideas. They have been around in a speculative form at least for over 30 years. In the present form there is very little design that goes on in the computer aided design system; mostly, the system is used for the representation of artifacts that have already been designed elsewhere. By the early 1960's the first CAD systems were starting to appear in prototype form. Their basic form has remained fairly constant since their conception, but what has happened over the last 25 to 30 years, basically, is that the systems have got smaller and smaller and cheaper and cheaper so that we are getting to the point where we can legitimately think of computer graphics as a mass medium. As a result of this growth, a significant market for computer graphics and computer aided design software has developed. This has motivated an enormous expansion in the effort directed at the development of the kinds of software that is needed to do something sensible with the machines. In addition, although much more slowly, there has been a process of what one might call the cultural absorption of the medium by good designers, as they slowly get their hands on these kinds of devices. The designers have begun to explore what really might be done with these systems, and how they can really start to design significant artifacts using this type of technology. This has been a much, much slower process and I suggest that it is because the technical capabilities we have at our disposal usually get very far out in front of our ability to do anything sensible with them. Doing something sensible with this new technology is one of the things I would like to talk about today.

I would like to spend some time talking about the techniques of representing architectural forms using computer graphics and geometric modeling techniques, and say something about what those various techniques of representation mean for the ways that we understand architectural form and for the ways we explore and
develop architectural ideas. The reason I think it is important, is that representation is, in a very important sense, the foundation of design. It seems to me, the way most important design processes work is something like this: the designer gets some kind of notion of what an artifact might be like, what a building might be like for example, and an idea emerges, the imagination throws up some sort of idea. Then that idea is represented using some medium. It is represented with pencil and paper, it is represented with a cardboard working model, it is represented by modeling clay, or it is represented by a computer aided design system. Then the representation suggests something back, the act of representation illuminates new aspects of the idea, tells the designer things about the artifact that he or she did not originally know, and suggests ways to develop that idea. The idea once transformed reflects back, the representation is changed and that in turn suggests something more.

There is an extremely complex engagement between the imagination and the representation of the idea that is related to the structure of the representation, to the way that the representation organizes information, and to the way that the information can be transformed. An architect who works developing an idea with clay, for instance, is going to take the idea in very different directions than if he or she works with a cardboard working model. One simply gets a different type of architecture. Different representation will suggest different things, and the kinds of operators that one has for developing and transforming that representation will start moving the idea in different directions. I would not characterize this as a type of crude determinism, where the physical capabilities of the medium determine what the designer can do, rather, I simply mean that the power of suggestion, the way the imagination catches fire and the way that the idea begins to develop is very different for different medium. I think most experienced designers know this very well—when they work with different medium, they will work in different ways, and
an experienced designer develops the skill of knowing when to change media, when to go from one medium to another. For a particular kind of design project, the designer chooses a particular kind of medium, and at different stages in the design project the designer knows when to go from, for example, soft pencil sketch on yellow paper, to hard line ink drawings, to cardboard working models, to polystyrene massing models, to calling in render in order to see what it looks like in light.

When a radically new medium that has different properties is introduced into design processes there is the potential for the design processes to operate in very different ways, and that enable the imagination to get into new territory. In fact, in the case of architecture, the introduction of a new medium opens up new realms of architectural possibilities and has the ability to carry architecture into a new domain. This is the real potential of computer graphics and of computer aided design technologies.

The realization of the potential that a new medium brings to a design field happens slowly. What typically happens when a new medium is being introduced into a culture is that it is used to replicate, maybe a little faster, and a little more easily, but replicate nonetheless. Early film, for instance, looks very much like filmed stage plays. It is as if someone pointed the camera at the proscenium arch, or carefully framed the proscenium arch with the frame of the camera, just cranked the handle, and put a play on film. Then, eventually someone comes along, a director or theoretician like Eisenstein who began to use close-ups and montage. It soon became apparent that film could be put together in an entirely different way, that it is not like a filmed stage play. A little bit later video appeared on the scene, and early work with that medium looked like film, until someone figured out that it is a very different sort of medium. With video, there is a small screen, with low resolution and it works in different kinds of ways. Instead of using big, stable, long
shots, one could go in very quickly and get close-ups. The video it was discovered, is edited in a very different way and, one worked with light and color in a different way. Eventually, something like MTV emerges, the result of a very different sort of medium than film.

This tendency to replicate an existing medium is a very general tendency and what we mostly see in computer graphics is exactly this kind of thing. If one looks for examples at most commercially popular kinds of graphic art systems, if one looks at the way they are being marketed and used, the two most popular features are paint systems and graphic systems that replicate existing media. In the case of paint systems even the name suggests an imitation of a previous medium. What a very successful paint system like MacPaints does, for example, is imitate as closely as possible the uses of the traditional hand medium. The metaphor is very clear and direct. Similarly, drafting systems replicate the processes of constructing two dimensional drawings out of vectors and straight lines, it replicates the type of things that one typically does with a parallel rule and compasses while working on a standard drawing.

I think we are at a point with computer graphics where we can perhaps begin to go beyond the replication stage. I am going to take some time to talk about various techniques that are available for modeling architectural form using a CAD system and explore a few of the properties of those techniques. Following that I will describe some of the work produced by students in my computer graphic classes and design studio over the last year. They have taken significant pieces of architecture and modeled those pieces of architecture in different way using the images that they produce to say something about the architecture; to make an architectural point. The reason I think this is a worthwhile way of looking at things is that one can discover the properties of a medium that are important, if one tries to do something significant with it. Little demonstrations do not get at the issues.
If one uses a CAD system to draft bathroom details, the important properties of the medium may be missed. So, what we have done is to take things like Palladio's villa and built representations of them. When this is done it does indeed begin to reveal the dimensions of this medium that would not be otherwise revealed.

Typically, a renaissance text, on architectural drawing and architectural design introduces the topic of architectural drawing and architectural design by introducing the drawing instruments; the architect's tool. This was a predictable way for these texts to get started laying the foundations of architectural design. The renaissance text would begin by introducing the straight edge, the square, the parallel rule, the compasses, and the dividers that are used to make euclidean geometric constructions such as parallels, perpendiculars and arches. These various constructions are exactly the capabilities that are provided by a 2-dimensional computer drafting system. In a very literal and direct way, computer based drafting systems replicate the functions of traditional architectural instruments and the construction of euclidean elements. This of course leads to a rather predictable design method.

Durand's treatise on architecture that appeared at the end of the 18th century is based on the technique of exploring architectural ideas through the medium of constructed line drawings. Durand lays out of the design method as the sequence of steps that he believed could be applied in the composition of any architectural project. To those of you with some background in software development, and software design, Durand's method is a very elegant example of a top-down structured design technique that works through a hierarchy of abstractions. It is the kind of progression that has been reinvented in software design in the 20th century. He beings by saying "start the project by laying out the principle axes of that project; a very abstracted skeleton of construction of lines that defines the basic organizing principles of the geometry." That is the first step. The second step
involves taking that very schematic skeleton and elaborating upon it by relating additional axes to the primary axis. So, one takes a layer of paper and lays that down on top of the first. Then using geometric construction, the secondary elements are placed in relation to primary elements. The third step involves taking that framework of construction lines, and laying out the walls relative to that skeleton. In the fourth step, the columns are layed out relative to that skeleton. The fifth step involves taking a vocabulary of elements, a menu of elements in the classical language of architecture, disposing them relative to the elaborated skeleton, and using menu selection and geometric transformations in order to accomplish this point.

What I have just described could be taken from the instruction manual for a standard 2-dimensional computer drafting system, with the notion of layering, the notion of working with a menu selection operation, the notion of transformation operations to put the composition together, and perhaps most importantly, the notion of recursive refinement as a way of developing a design. Starting with an abstraction and elaborating on that abstraction in order to build up the design, Durand has essentially outlined the combinatorial nature of this kind of design process. He generally takes the same construction skeleton and simply substituting within that structure different elements to produce a range of variations. What has been developed is a representation method, or in other words, a way of thinking about a medium and the use of that medium to represent architectural form. Then within the framework of that representational method, a way of thinking about architecture emerges. The theory of architecture is developed into a working method in a very clear and direct manner.

Anyone with a computational background will recognize that what I have described as the five steps taken in the design of building in the classical style, is the way a formal grammar works. It is possible to describe these five steps in other
words, where one begins with a stopping symbol and some production operations that pertain to it. The productions are applied and the stopping symbol then gets rewritten as a stream of symbols. Additional rules are applied, the symbols get rewritten, a process that continues recursively until a legal sentence in that language emerges. When the mathematics of the process are elaborated it becomes possible to write a grammar of architectural forms, an architectural grammar that can be defined in a precise and vigorous way.

Somebody raised the question yesterday, concerning whether it might be possible to write an algorithm to generate classical architecture. I have proof that such an algorithm can be written. About ten years ago, one of my colleagues and I wrote a grammar of this type that generated Palladio’s villas. It is a concise and short grammar that generates all the villas that Palladio designed, and all of the villas that he produced, as well as many 1000’s of fake Palladian villas that were very convincing. Perhaps most interestingly, it was discovered in recent years, that some villas designed by Palladio that had come to live in the Royal Institute of British Architecture drawing collections turned out to be designs that were generated by our grammar. So, this sort of thing is possible, it can be done. It is not particularly difficult and it is not limited to classical architecture. For example, a couple of my students wrote a very elegant grammar that generates Frank Lloyd Wright’s prairie houses in a similar kind of way. A grammar has been developed for traditional Japanese tea houses, as well as for a variety of other types of compositions. Thus, the notion that architectural composition can in fact be generated from a fairly simple grammar is demonstrably true.

Using the capabilities of the CAD, it is possible to construct a piece of classical architecture such as one of Palladio’s villas, step by step using a constructive technique that is directly analogous to the method Durand laid out. The only appreciable difference is that computer graphics are used instead of rulers and
compasses. One simply begins with a cross axes as Durand suggested, and then elementary geometric figures are related to the skeleton until the skeleton has been elaborated. The skeleton is elaborated step by step and the lines are built up using elementary geometric figures related in fairly simple ways using constructive capabilities such as circle, arch and vector fitting. Finally, when the skeleton is elaborated, the detail forms of the plane construction are substituted and the whole composition comes together.

If one were to construct a classical piece of architecture, such as one of Palladio’s villas, line by line, arch by arch, it would clearly be a worthless kind of process. If, however, one structurally understands the composition, thinking of it in terms of the way that the geometry builds up, the way that the vocabulary works and the way that the syntax works, then the entire composition can be understood in terms of the vocabulary syntax, and geometric transformation. There is a distinct and elegant way of putting the composition together, and always, there is the notion of a structure that is being formed.

It is possible of course, to go a step further and think of a building, represented not as a set of lines on a 2-dimensional surface, but as a set of lines in a 3-dimensional Cartesian system. When one starts to do that, it becomes necessary to introduce some additional operation for putting the forms together. The operations for going from a 2-dimensional construction plane to a 3-dimensional array of lines in space, as it turns out, can be thought of as ways of structuring architectural thought. One of the operations, for example, involves taking a plan, developing it in a construction plane and then using a project or an extrude operation, to simply develop that 2-dimensional form along a third axis in order to create a 3-dimensional spatial organization. This operation is very elementary to implement, and any kind of simple 3-dimensional CAD system typically has the capability. Actually, this operation can be very dangerous, in so far as, I have spent
years and years teaching first year architecture students not to start thinking about architectural form as a nasty little plan that is popped into 3-dimensions by extrusion, and then along comes computer aided design systems that give one precisely that sort of capability. Nevertheless, extrusion is a legitimate way of moving from developing 2-dimensional surface structures into 3-dimensional forms. The point I was trying to stress, however, was that this is not all there is to architecture. The operation of taking a profile and evolving it into a 3-dimensional form is not only a technique for the input of form, but it is a constructive technique, a way of putting together a formal world that the architect can explore.

The design idea is that one can define a profile and then by moving the form along a given axis, generate a 3-dimensional form. A variation of this involves constructing a profile in a 2-dimensional construction plane, and then rotating it around an axis instead of extruding along an axis. The result could be a dome in one of Palladio’s rotundas. Both elementary ideas, but very important ones in the history of architecture. The idea of starting with a profile is preserved, but a different constructive operation is used to put the form together in 3-dimensional space, and a different world of forms starts to emerge. This may sound trivial, but I suggest that it is a very illuminating way of understanding classical architecture. Simply take the notions of a profile, extrude it along an axis or rotate it about an axis, and it becomes possible to see that a tremendous amount of what classical architects have done comes out of those simple ideas. It is the way that the formal worlds that they explore are structured.

This 3-dimensional exploration of the world leads rather naturally to the notion of wire-framed modeling. Serlio, one of the great renaissance architects described the projection operation as a constructive operation. He also described, in a very interesting way, the advantages and disadvantages of both wire frame representation and hidden surface representation. There was a renaissance
conception that is sort of amusing in today’s terms, which argues that the wire representation may in fact, be considerably superior to hidden surface representation because it revealed what the eye of God alone can see. In fact, this is the great quality of wire frame representation, the ability to reveal relations between interior and exterior surfaces. When form is thought of as a set of lines in space, the architecture is seen as discontinuities and edges, abstracting away nearly everything else. In contrast, hidden surface representations can be thought of in a more directly visual and less abstract sense.

Surfaces, opaque surfaces, surface that obscure other surfaces and surfaces that modulate light, leads one to start thinking about architecture as a set of surfaces in light. This relates to Alberti’s definition of architectural design. Alberti’s defined architectural design as the correct and exact, joining together of lines and angles that compose and form the surfaces of buildings. Surfaces in light is a different way of looking at architecture; the two ways are not exclusive of each other but they are looking at different aspects of architecture.

Surface modeling can clearly be done with a computer graphics system, but it becomes necessary to know where the surfaces are, so more information needs to be added to the representation. It is possible to take a wire frame representation, define the way lines bound surfaces and specify surface properties sufficiently well to allow a hidden surface drawing to be produced.

The third representational technique to be considered is one that uses polygons as the primary building block or primitive. Polygons can have color or tone, and the process of composition is not just a process of relating lines and arches to each other, but a process of relating the the sizes, shapes and colors of polygons to each other. The use of polygons as a primitive opens up a different way of thinking about graphic representation. This difference in terms of traditional medium, correspond to the difference between line drawing and painting. The first involves
the manipulation of lines and angles, and the second the manipulation of color and shading. As soon as one begins to work with polygons, it opens up different ways of constructing pictorial space in a 2-dimensional image. The architect is forced to deal directly with issues of space, surfaces and light as he or she relate polygons to each other, and this technique of graphic representation is another way of thinking about architecture.

Architectural form, construed of as a set of surfaces in light and the manipulation of those surfaces, is very much like working with a cardboard working model in which cardboard polygons are constructed and fastened together to make 3-dimensional forms. In order to achieve this with a CAD system one need simply write an algorithm to render the solid surface and lighting effects that are desired.

The simplest way to do that is to use cosine law shading and assume a monocromatic surface. With diffuse directional light, the application of Lombar’s law indicates that the intensity of light reflected from any given surface is related to the cosine of the angle of insistence. Lombar’s law states that angular information is encoded, or more precisely specified by the intensity of the light reflected from the surface. There is because of this property more information in this type of representation than there is in a wire frame representation, or hidden line perspective.

Neither the wire frame representation, nor the hidden line representation convey the angular information to the same extent as a polygon modeling technique. Using the polygon modeling technique, one is able to take rather complex composition and render the spatial qualities rather effectively. This polygon modeling technique is related to the traditional architectural technique of making a massing model out of wood or polystyrene, coating it in 50% mat gray, taking it into the photograhics studio, illuminating it very carefully with directional diffused light, and producing a photographic image. This is a fairly
traditional architectural modeling technique that is used to produce abstract representation that show surface qualities and emphasize the massing of the form. The process of constructing a model out of wood on polystyrene, however, is rather elaborate, so the behavior of light on curved surfaces has traditionally been explored using techniques like watercolor wash and charcoal rendering. Both of these techniques, the watercolor wash and charcoal drawing, are especially sensitive to the modulation of curved surfaces. Using either these traditional techniques or the polygon modeling technique makes it possible to start thinking about architectural form as surfaces in light.

Various kinds of lighting effects can be explored as architectural issues, by elaborating the description of light and the description of the surface reflecting that light. Interpolation for instance, is a technique that is a directly analogues to taking a charcoal drawing and smearing it with one's fingers across the surface, or taking a watercolor wash and braying it in order to produce shading. The technique of rendering a model using smooth shading begins to show something that is essential about architecture, namely, the way light is modulated by different types of surfaces, and the tonality that is experienced when the artifacts are produced in real light.

Computer aided design as a medium provides the architect with an opportunity to explore the properties of simple surfaces in light and the ability to understand those properties in the simplest possible way. It is of course also possible to elaborate the way in which the surfaces are explored. By defining more surface information and defining a more elaborate lighting model one can begin to explore the architectural issue concerning how an abstract form can be materialized in a variety of different ways. The materialization of architecture is the kind of thing that one generally explores in the design development stage. However, it is quite clear that if one takes the same form and materializes it in different ways the form
acts in a very different fashion. Thus, a manipulation of surface properties is not simply a question of rendering, gratuitously, something that has been designed, but rather it is a way of exploring designing ideas.

One way to think of architecture is to think of it as beginning with an abstract geometric form and then explore ways to materialize those forms with real material in real light and computer graphics is the sort of medium that gives one a way to do that. Design development can be explored in a very interesting way using surface modeling techniques. It is also possible to combine wire frame and surface modeling in some interesting ways, such as representing the exterior surface of the building in wire frame and representing the interior surfaces in shaded surface modeling. Using this technique one can begin to understand the relationship between interior and exterior surfaces. It is also an elegant way of showing the relationship between the planned form, the plan of the organizer and the way that plan develops into a 3-dimensional object.

I have talked about representing buildings as a set of lines on the 2-dimensional surface, as a set of lines in 3-dimensional space, and as a set of surfaces in light, a fourth way of looking at architecture is as a set of closed solids in space. This method of representation, in other words, is architecture as volumetric composition. Solid modeling is architecture where one begins thinking about the construction of architectural forms as a set of 3-dimensional building blocks, where those building blocks are put together in space in order to define the building's volume. One can do that with a polystyrene or wooden model, but the endeavor becomes more interesting when one begins to think of the solids as being put together in more complex ways. This is not merely the idea of graphically representing building blocks, but the idea of using some fairly sophisticated combination operations to construct the artifact.
Le Corbusier explored the idea of architecture as volumetric composition. He introduced a vocabulary of closed solids: a rectangle, a prism, and a sphere, and then demonstrated how a diversity of form could be generated by starting with a few elementary closed solids, and then putting the closed solids together in complex ways. Complexity emerges out of simplicity, and the poetry of architecture emerges from the set of simple forms when geometric transformations are applied in order to combine the pieces. What this process has become in the computer aided design world, is the technique of solid modeling. The designer begins in exactly the way Le Corbusier described, with closed 3-dimensional solids as the basic architectural vocabulary and an array of the combination operation. Cubes, spheres and tetrahedrons are very simple things, but complexity of form can emerge very quickly as a result of the constructive operation. From a very simple vocabulary, and the introduction of a few constructive operations, one gets a universe to explore, a universe of design possibilities, which are in fact, highly structured and rule-like in nature. To illustrate how one can get some new insights into architectural form by thinking in terms of solids, imagine a sphere and a rectangular box, place this box on the equator of the sphere and save what results from the intersection. The form that results from the intersection of these two elementary objects, is a form that has had enormous importance in the development of Western architecture; what one gets from the insection is a traditional architectural form in Byzantine and Ottoman architecture. It is a transitional form lying between the geometry of squares and the geometry of circles.

Using wire frame representation to understand architecture one can develop the same shapes, or one can develop them with a single operation when thinking in terms of a volumetric composition. Solid modeling is a very interesting way of understanding architectural form, a logic based upon using union, intersection and
subtraction operations and a small set of elementary shapes as the vocabulary. Solid modeling in the most direct sense, is a composition of spaces. It is a very interesting way of thinking about architecture because it focuses on the point that the spaces of architecture are actually inhabitable.

What I have done this morning is discuss various methods of representation and the influence that the form of representation has on the emerging architecture. The different techniques of representation suggests different ways of understanding architectural form, and the different ways of understanding architectural form, together with the associated operation, provide different ways of developing and exploring architecture.

Basically one can think about architecture in terms of a computational background by beginning with the notion of primitives, graphic or spatial primitives. What I have done during this presentation is introduce different ways of thinking about architecture that are based on different systems of primitive.

Architecture can be thought of as a composition of lines in spaces, lines on a 2-dimensional surface or lines in 3-dimensional space. In the simplest case, the primitive can be a vector, a straight line segment that can be elaborated in order to deal with arches and spline curves. Another approach involves using surfaces as the primitives for constructing architectural compositions. The easiest way to do this would be to take a 2-dimensional polygon, bounded by straight lines and build up everything from an array of flat surfaces. A much more complicated vocabulary of primitives would be the set including surfaces that are defined by spline edges. The final approach I have considered involves primitives that are solids, and thinking of architecture as a volumetric composition with a vocabulary that includes cubes, spheres, cylinders and cones.

These different classes of primitives relate to different definitions of architecture and reveal different aspects of the architecture. Alberti, as I have pointed out, gave
a classical definition of architecture, defining it as line modeling. He claimed that lines are the primitive and architecture is the composition of lines and the angles created by their intersection. Le Corbusier defined architecture in another fashion, as the masterly, correct and magnificent play of mass in light. It is possible to relate both of these very traditional notions of architecture, in a formal sense, to a choice of primitives that once chosen, determines the structure and exploration of architecture that follows.

In addition to primitives, what is needed in the formal computational sense, is a set of operations to make the composition. In traditional architectural techniques, the operations are considered to be such things as drawing a line with a parallel rule, constructing a perpendicular, or striking an arch through three points. These are the tools, that in other words, can be applied to the primitives in order to construct a more complex structure. For the technique of drawing, a traditional way to define a universe of architectural form, exploring that universe and constructing artifacts involves taking a particular class of primitives and the set of drawing instruments that are used to work with the primitives. In a computer aided design system there are also certain defined primitives and a set of operations that are available for manipulating those primitives. It is possible to take the primitives and the available operations and put them together and get, as a result, a universe of form possibilities that are defined by the application of the operators to the primitives. In fact, what emerges in mathematical terms, is an algebra. When a set of primitives are defined and a group of operators are applied the resulting forms are going to have some sort of algebraic structure. A perfect example of the formal properties of architectural form occurs with case of solid modeling where the universe being explored has the structure of a Boolean algebra.
Wilcox: I am curious about something. The structure that you have described so far is basically a Euclidean geometric system, where one begins with primitives that are defined within a Euclidean system. Has there been any attempt to define a system using topological geometry rather than Euclidean as a starting framework?

Mitchell: Yes, I for example, started with lines, but it is possible to explore the domain in many other ways. Architecture, for example, can be conceived of as a collection of spaces and the concern with the adjacency of those spaces.

Wilcox: What you have presented then are just a few among a whole universe of possible representational devices.

Mitchell: Absolutely, and that gets at the important point that I have been trying to make which is, the choice of representation determines the form of the idea, the nature of the exploration, and the type of universe that is going to be explored.

Wilcox: What you have said also relates to the earlier discussion concerning complexity and simplicity. Using one set of primitives means that simplicity is one thing, and using another set means it may be something else. For example, when solids are the primitives, lines start becoming a complex phenomenon, and when lines are the primitives, lines are simple and solids become complex.

Mitchell: One of the very important things in the design process is the kind of decision making that leads to one form of representation being used as opposed to another. During the process of going from one set of representations to another set of representations some architectural decision gets made. For example, within the framework of wire frame representations one could move to a surface representation. Clearly, there is a lot of decision making that occurs at that point, a decision has to be made as to where to put the surfaces and additional decisions
have to be made regarding the various surface properties and the various relationships between the surface and light.

There has been a tendency in the early days of computer aided design, to think that one could construct a computer-aided design system that has one set of representation, or one way of representing a building and the means to edit that representation. It has been shown, however, that that belief is very naive, it does not work. It is very naive because it eliminates one thing that is absolutely crucial in design, the process of changing representations. A change of representation reveals different things, it brings different issues to life, and in fact, it is a way that design decision making occurs. Everybody who designs knows this. As one moves in architecture, for instance, from a soft pencil line drawing on yellow paper to a hard lining plan, decisions are made, angles are defined and dimensioning conflicts are resolved. As the hard line drawing is transformed into a card board working model, again, a lot of decisions are made.

Some CAD systems now provide the capability to working in a bit map with a paint system; the net result being an unstructured representation. So the system does support, in a rather primitive way, the ability to change representations. With an AI application, it might be possible to start writing algorithms to enact the change of representation. It would certainly be hard, but it gets at the heart of architecture.

Cavalier: With your system is it possible to come down into a building and actually move through the structure that has been drawn, in essence, come in one door and go out the other, or in other words, wander around?

Mitchell: Certainly the principles for doing that are straight forward and very well understood, it is simply a matter of machine cycles. Right now, computer aided design is mostly thought of as a novel way of making static images. However,
because of the inherent nature of computer graphics images, it is possible, not only to move the viewpoint in a 3-dimensional model, but the potential also exists for real time transformations and real time parametric variation of the object. As processing gets cheaper and cheaper, I think people will start to think about the drawing as a dynamic object. That is a fundamental change. That is a bigger change in the way that we think of drawings than was the Renaissance development of constructive perspective.

Cavalier: The difference in time between rendering an artifact on a CAD system and doing it by hand must be enormous.

Mitchell: That is a complex question because the amount of time it takes to construct a model really has little to do with the capabilities of the technology. As long as one has the basic capabilities it is possible to efficiently construct computer models. As long as one has a structural understanding of the building, an understanding of the vocabulary, the syntax, and the way the thing is put together, it is possible to define a very short sequence of moves that will get one to where he or she wants to be, using a CAD system. If however, one were to start constructing line by line, surface by surface, in a kind of match stick modeling technique, it would take years to put the image together. It is not a question of technology as much, as it is a question of understanding the architecture, of mapping the architectural idea onto the algebra.

People sometimes ask me, “as a strategy for teaching architecture students, should one teach the students who want to design on a CAD system programming?” The answer that I usually give to them, is that the use of any kind of CAD system is in fact programming. What one does is define a sequence of commands that get executed. The sequence may not be in a written command language, it may be done with gestures, there are a lot of different ways to
construct an interface. However one does it, a sequence of instructions are given to the machine, and the machine executes those instructions to put together a model. In a language like Pascal, a sequence of instructions get compiled and then executed in a way that we are familiar with when thinking of computers. In an interactive CAD system the commands get interpreted and executed one by one, but there is still a program. The issue in programming is always to make intelligent use of abstraction, to get a structural understanding of what is being done, and express what needs to be expressed in as a concise and elegant way as possible.

As a definition of understanding something, I would suggest, that one really understand something if he or she can write a concise and elegant program to generate it. There is always a long complicated and messy program for generating something, that is trivial. One can always make a drawing line by line. However, if one can take an apparently complex drawing and understand the logic of that drawing so well so that it can be executed in a few elegant commands, we have as an image, the Zen painter. We have as an ideal the Zen painter, who after looking at the bamboo tree for six months, captures it on paper with one elegant stroke and then walks away. The point I am trying to make is that once a reasonably adequate computer system has been defined for rendering architectural models, we come back to the question of architectural understanding. It is not a matter of operator competence. The best people I have seen working on the CAD are architects and designers, and they are not necessarily the most proficient operators. They may be dreadful operators; they may stumble around, crash the system, all sorts of things; what matters is the level of architectural understanding.

H. van Dijk: Do you feel that with this new technology you could deal with issues that you could not deal with using a manual system of representation?
Mitchell: Absolutely. In a simple way, what happens is exactly what one would expect, the students are able to very rapidly explore a much wider domain of ideas. One of the typical problems of teaching architectural students is that even if the students are good, their facility for representation is not such that they can draw complex ideas very rapidly. Traditionally, the student develops a few alternatives, but he or she would not be capable of exploring a large domain of possibilities. Give the students additional representational facilities and then design becomes what it really is, the exploration of a broad domain and the focused development of something within that domain.

The second thing that happens with a machine based representation, is that analyses of that representation can be performed easily. If one has a solid model representation, for example, that is geometrically complete, it is possible to assign material properties to the pieces of the model and perform an engineering analyses of it. So, not only is it possible to explore more possibilities, but they can be more thoroughly analyzed. The design is not just a speculation of how something might work, it is a demonstration of how it will work.

If we go back into the Italian Renaissance, we find that a lot of our ideas about design, that we take as given, emerged at that point in time as people were trying to figure out ways of handling the increasing complexity. Two different ideas about design emerged at about the same time. On the one hand, Alberti developed the foundations of the idea of design by drawing, a tradition within which architecture evolved. Galileo, on the other hand was saying that the thing as seen cannot be believed. It is possible to derive a mathematical formula and design by manipulating the symbolic model. We get as a result two different approaches to design. One is the exploration of ideas by making marks, by drawing, and the other is the exploration of idea by manipulation of the symbolic model. Engineering, by
and large, has followed the symbolic modeling approach and elaborated on the notion of being able to mathematically model artifacts. Architecture has very strongly taken the root of design by drawing.

There are obviously connections between the two approaches. In architecture schools, we try desperately to teach students structure and give them some notion of engineering, and they hate it, because working with mathematical models is alien to the tradition of architecture. In many engineering schools, the tradition of design by drawing has almost completely died with very, very little of it occurring today. There has been an enormous divergence in the application of these two approaches over the years. However with computer aided design it becomes possible to bring about a convergence of the two approaches. It is possible to operate an analysis on, and generate an image of the same representation. There is a potential to remove the enormous divergence in design approaches that has occurred since the Renaissance.

There is another thing that happens with the CAD system that has some practical significance. There is, as a result of using a CAD system, a devaluation of the image. In the field of architecture, traditionally, a good image carries a lot of conviction because it is hard to make, and people have to put a lot of their heart and soul into it. Often a lot of bad projects get accorded a lot more credibility than they might otherwise have just because they are well drawn. What happens with computer graphic systems is that they democratize design. Now, not only can anybody produce good images if they think about it, but they can do it very, very fast. Interestingly, all of those issues of representation get factored right out and one begins to have to look at the architecture. I find that a very positive kind of thing.

Davies: Have you explored modeling that involves a specific site and place, a specific form and how it relates to sunrise, sunset, interior, exterior, and so forth?
Mitchell: I have not done much with that yet. However, in attempting such an exploration there are two levels at which one has to think about it. One level is that of the theoretical framework that is needed for doing that kind of thing. The other level can be thought of in terms of the practical computer technology that is available to do this sort of serious design work, and that, I am afraid, is a bit beyond us in terms of the cost, complexity of the software, speed of the machine cycles and the amount of available memory. To model a complex context in sufficient detail at the moment is a fairly complicated issue. I do not see any major theoretical difficulties in doing that. For example, it is a simple algorithm that is needed to calculate how to cast shadows, and to calculate how they move as the sun moves across the sky. If one wanted to do something like taking a building, orienting it correctly, and then simulating the specific site, it is theoretically straightforward, but there is a practical limitation.

What I have been doing during the past couple of years is focusing on building as an isolated artifact. The representational style that follows from this focus involves presenting the buildings as objects floating in black space. I obviously do not think that this is an adequate general view of architecture, it is simply an intellectual strategy that I have used in order to get started with the exploration of the issues.

Nasar: The CAD system that you have described is obviously a powerful tool, but I have a concern that I want to raise. This system may lead some architects down a dangerous path that they seem to be naturally moving toward. A dangerous path which involves a concern for geometry, independent of what the building is actually for, which is namely, the inhabitants. I raise this as a concern because I think that it should be recognized as a problem that may be amplified with this new technology if certain tremendous potential are left unconsidered. One is the possibility of using a CAD system as a means of interacting with a client. It should be possible to tailor
the design to meet the client's needs on line. The other possibility involves actually simulating patterns of behavior in space, and then designing the building around the patterns, creating a framework that supports rather than ignores or worse yet, thwarts the intended behavior. I am very concerned looking at some of the trends that are occurring in architecture right now, it seems that geometry is frequently more important than people.

Mitchell: That is something that concerns me a great deal and is one of the reasons why I emphasized issues, such as, model the inhabited space first, and then construct the building around the inhabited space. I think that most bad architecture, and I mean bad in the sense that it fails to consider the inhabitants, comes from the failure of the imagination. The architect did not understand what the building was really going to be like to those who inhabit the spaces. The failure of the imagination occurs, I think, for two reasons. One is the traditional techniques of representation such as plan elevations and small scaled models. Architects think they have pretty good spatial imaginations, but it is evident that it is not as good as they think it is, and the techniques of representation are simply not adequate to compensate for the deficiency in spatial imagination. There is often a simple kind of failure on the part of the architect, a failure to really know what the designed space is going to be like. It is not the case with very good architects, but it is the case often enough. The other reason for a failure is that there is often just not enough time. If some sort of crude measure was made of the amount of care per square foot that a 19th century architecture was able to put into a building as opposed to what a typical 20th century architecture is able to do, I think it would reveal that the present day architect is able to put in significantly less. If, however, the representational facility of the architect is increased, I think his or her capabilities to be able to understand will also increased. The ability to see and
understand, the ability to project oneself into what is being designed, is, in the end, one of the best safeguards against a barren and abstracted view of architecture. Rather than taking one away from the physical reality of what the buildings are going to be like, strangely, computer graphics can bring one much closer to the architecture.

Let me return for a moment to the very interesting issue of how architectural knowledge gets embedded in a computer aided design system. Even at the lowest level, the level of primitives and operators, we see that a substantial bit of architectural knowledge is already present. For example, the operators in a standard two-dimensional drafting system embody knowledge of Euclid's elements and geometric construction. How a parallel is put together, a perpendicular is made, how arches are constructed, and so on. If one were to look back to medieval architecture, the content, what one learned to be a master builder, or architect, was basically the knowledge of Euclidean geometric construction. Euclid's elements were the foundation of architecture. We may not think of this as embodying a whole lot of architectural knowledge, but basically when the algebra is put together, the primitive and the operators, a fairly substantial amount of knowledge is embodied in the system. The primitives and the set of operators defines the universe of architectural possibilities within the algebra, but there is too much in that universe. It is a big and complicated thing and there is all sorts of garbage in there that is of no architectural interest. One wants a smaller and more structured universe of possibilities to explore, so it becomes necessary to start looking at a relevant subset in that universe. In a formal computational sense, the way to specify a subset within the framework of the existing algebra, is to define a production system, or what is in fact, a grammar. With a specific reference to architecture, what one might choose to define is a grammar that operates in a
universe of lines, surfaces or solids. What is obtained is a set of rules that basically determines how the various elements get put together.

The Palladian grammar, that we wrote, for instance, embodies a lot of knowledge about the Palladian style of architecture. It is possible to use that grammar, and create Palladian architecture very convincingly, thus, clearly there is some non-trivial knowledge encoded in the system. What that knowledge is, is a set of rules that operate within the representation, representing plans as a set of lines and defining some production sets. A design system that writes a grammar, describing how things get put together is a significant step along the road to intelligent design machines. Once the grammar has been written it becomes possible to do several other things. Once the language of architectural form is described in purely formal terms, it becomes possible to start thinking about the issue of the semantics of this language which is a large, complicated, and still very ill understood thing.

I do not want to speculate too far because it is going to take a lot of theoretical work to get semantics into the language. It will be a major intellectual project to say the least. There is a theoretical level and then once theory has been worked out, there is the level of implementing the theory in a practical computing systems. There is a lot of work to be done along those lines, but I think it is the sort of direction that is important to pursue. I think rather than making a fundamental distinction between representation systems and design systems, it is better to think of it as a hierarchy of systems, with low level systems that operate close to the level of primitives and do not embody a lot of knowledge, to higher and higher level systems that involve more and more architectural knowledge. The further one moves up that hierarchy, the more a given system starts to look like a real design systems and less like a little editing systems.
Dilnot: How does a system such as the one you have described today cope with an architect like Gaudi?

Mitchell: Gaudi is very interesting, and his style of architecture is very interesting in terms of a computational framework. He worked with analogical computation instead of digital computation, but still, a very well defined computational framework.

Dilnot: Well, I tried to think of the most difficult architectural style, and Gaudi was the best I could come up with.

Mitchell: Let's try to analyze Gaudi in terms of the computational framework that I have laid out. I think it will make a very good example, because Gaudi worked within a very well defined computational framework. The basic primitive that he used was a chain hanging from a couple of points. The way he manipulated that primitive was by hanging weights on the chain in order to form a catenary curve. Gaudi defined the catenary curve as a primitive, instead of a straight line. A line can be defined by six parameters, three coordinates for each end point. The catenary curve is more complicated, but it is still a parametric primitive in the same kind of way. Gaudi literally worked with chains and weights as his model. He defined forms by hanging various catenary curves in space and then covered them with cheesecloth when he got them to the shape that he wanted. He would then take photographs of the structure and turn the photographs upside down in order to achieve a very complex vaulting form. The computational form of the catenary curves is really diabolical, involving transcendental functions, but the analog is very easy. Analog still has advantages, even today; in the case of constructing catenary curves, one just hangs up the chains and computes various curves by moving the weights. It also makes a great deal of structural sense, because it not only gives one
the form, but it also solves the structural problems in so far as the compression encountered in the vault is proportional to the tension in the chains.

One can see very clearly in Gaudi's framework that there is a primitive, a 2-dimensional form, the operations for transforming it into a 3-dimensional form, and that it combines both the analysis and the development of the geometry. It would be fairly hard to reproduce Gaudi's work within the framework of a CAD system that is primarily intended for work with straight lines and arches and the kinds of curved surfaces that emerge from that sort of geometry. It should be clear, however, that once the architecture is understood in the sorts of computational terms that I have been describing, a kind of general approach follows. Gaudi's is not the same kind of architecture, so one would not expect to see the same kind of CAD system, but the same sort of general approach still applies.

Ettinger: When one does this kind of analysis of Renaissance architectural form the relationship that exists between the resulting representation and the original structure is quite similar because of underlying parallels. However, when this kind of analysis is used with Gaudi's architecture, the relationship that exists between the representation and the qualitative experience of the actual structure is quite different.

Mitchell: The qualitative experience may be occurring at another level, and I have been dealing with only the basic definition of the form. There are certainly other levels to Gaudi's architecture, without question. There is for example, the issue of how the surfaces are modulated by light and how they are materialized and how they are decorated. These are other levels of the architectural composition not captured in a study of form.

Making that connection between a method of representation, and design intention is non-trivial. The reason that something like solid modeling is good for
Renaissance architecture is based on very sound theoretical principles. The Renaissance architects were concerned with volume and symmetrical relations, and these sorts of things can be naturally explored within this kind of representation system.

Ettinger: Do you think it is possible to explore the development of the union between the method of representation and the final form of the artifact by forcing the work on non-related architectural forms?

Mitchell: Yes, absolutely. It is simply a matter of inventing a new form or universe of forms to explore. Architecture is a cultural art, it is an art that is deeply embedded in the specific cultural and historical consciousness. It is likely to be true of all art, but it is very evident in architecture because of the way buildings exist over a time. I think the way to understand the union is to begin to understand how a historical tradition engages a new technology and the way it is transformed by it.

Dilnot: There is an important point that needs to be made about modes of representation. A watercolor rendering, for example, is what one might refer to as a subjective representation. Yet, in this representation one finds a complex connotative structural analysis. I suspect that the building has been transformed in order to present its structural state. A photograph of that same building may show us the building more effectively, but it shows us only one angle of the building. The computer representation shows us certain things, but clearly at the same time, does not show other things. One constantly gets the sense that designers have failed to acknowledge that the various modes of representation are simply modes of representation that contain both possibilities and absences.

Mitchell: Absolutely. You are accentuating what I think is the central point in what I have been saying concerning the nature of representations.
Dilnot: There is a sense in which this CAD technology builds in so much complexity that I begin to wonder if it is really worth the colossal investment of time and money, given the various other modes of representation that designers already have at their disposal?

Mitchell: There are answers to that question at several different levels. To answer at the first level, let me go back to your premise which I think is very important, that is that a representation is an abstraction, a selection and a re-reading of the original. Whenever a person makes a representation of something, he or she is making an interpretation, the person is deciding what is important in the thing. That is why designers are taught to draw. It is not simply a matter of being able to put the lines in the right place, but it really is an issue of how one looks at the things, how one analyzes them, how one understands them, and what one decides is important, and what one decides is unimportant. In architecture, as well as the other visual arts, there has always been a central belief that representation is interpretation and interpretation involves simultaneously, two sides of the coin, accentuating some things and getting rid of other things. Now, what that means, regarding the question of whether CAD systems are worth it, is that absolutely, they are worth it. I do not think the solution is to say: "computers are cheap and powerful and we can develop techniques for representing just about anything we want, as we keep adding and adding to their power, while building more and more complex machines." I do not think it is ever possible to escape the fundamental artistic question which is, "what is it that we are interested in, how do we want to structure the exploration that we are going to carry out?" The investment becomes worth it if the representational technique that has been developed opens up a domain of architectural exploration that is in some sense a non-trivial domain. If a representational technique that is developed opens up realms of speculation that
are useful and important then my answer is, "yes it is worth it". Gaudi's work is a good demonstration, for here we see a representational technique for exploring architecture that could obviously not been done when drawing with a parallel rule and triangle. So again, the answer is yes, because the computer aided design medium is sufficiently new and sufficiently powerful that it is at least possible to define new domains that are both useful and important. I have to say that it is an act of faith, because I have yet to see it demonstrated, but the potential is there.

Sears: I think it is possible to add to the argument for the importance of computer aided design systems, for as was mentioned earlier, there is the potential to use one of these systems to transfer from one medium to the next, and that process of transfer is the design process.

S. van Dijk: I just want to add one thing to the discussion concerning the representation as the biggest problem facing architecture. Perhaps, not too far in the future, we will have a solution to the representation problem that can be borrowed from the movies. If we take what Todd Cavalier has mentioned about moving through the architectural space to be important, I think it would be possible to use a CAD system to create what amounts to a cinema in the round. It should be possible to put the architectural wire structure renderings or solid model rendering in a life-size round situation, thereby making it possible for the client to get a sense of the space in the final solution, because we have increased the scale and put the structure either all the way around or half-way around the observer.

Mitchell: I agree up to a point, if the issue that the architect wants to explore, the issue that he or she wants to present to the client, is the optical experience. If one wants to focus on that and explore those sorts of issues, then a technique like that is very important. However, if one wants to explore the notion of how activities are
organized by a plan, there are a lot of other ways to do that without bringing in all the irrelevant issues that are likely to arise. What I am getting at is, that a representation has to be connected to the precise architectural intention, and to the issue that is being explored. One has to be careful not to introduce irrelevant things into the representation.

S. van Dijk: Is it really that irrelevant?

Mitchell: Yes, I think so. There is an unspoken presumption that the more optical effects that can be rendered in an image, the better the image becomes. It is very much the case with architecture, and I think all architects know, the very precise watercolor rendering that one does, comes at the end of the job when one is trying to show what the thing will look like. A design media is a much more abstract representation. Because it is impossible to deal with the full complexity of design all the time, the complexity is reduced, or abstracted so that specific issues can be examined. A plan is an astonishing abstraction that reduces all the complexity of a building to a few lines on a plane, a wire frame image is an astonishing abstraction that factors away everything to do with light. A life-size, or an architectural scale rendering is not an appropriate design medium for architecture. So what I am getting at is that there certainly is a role for immensely realistic optical presentations of the buildings, but it is only one tiny aspects of architecture. I think your point is valid, and that it is an interesting and exciting potential for architecture, but I would also defend the proposition that very simple, abstract representation have a clear utility, and in fact are much more appropriate for architecture.

Novak: A representation does not necessarily need to be an abstraction that reduces information, because in the case of the computer graphics there is the
capacity for recursive refinement and the capability of creating information. Ornamentation, for example, which has dropped out of architecture, is highly recursive and could very well be put it back in with one fell swoop.

Nadin: I would like to say a few words about the issue of representation. I can say from reading some of the very recent writings on this issue, that there is a crisis in our understanding of representation. One will find as I have, that lots of people have started asking very critical questions about representation. They are more than ever, aware that in the process of re-presentation, agreeing here with Clive Dilnot when he said, representation is a re-presentation, that we are also constituting things. We are not only re-presenting things, but also constituting things and the two components are quite important.
PART VI

Design as Knowledge-Based Problem Solving Activity

Discussion led by B. Chandrasekaran
Chandra: Let me provide an outline of what I am going to be talking about. Design is a very complicated activity, and my interest in design is as a person who is interested in the computational theory of intelligence. I think design is something that minds do. If that is true, then one goal can be stated as, what is design, that minds can do it? In other words, I would like to make the connection between the somewhat more generic structures of the mind and the particular activity of design. At the same time, it must be noted that design is not a particular activity, design is a generic activity because there is something common to the activity of designing buildings, electronic circuits, engineering objects, or computer programs. So, this is really an attempt to link two kinds of generic things; one is generic mental structure, and the other is a generic task called design. The question is, what is the mind that it can do to design?

I think by now, many people have probably been exposed to the notion that the current AI paradigm is one of trying to understand mind as computation. The idea being that it is possible to conceive of the mind as an information processing device. The mind takes information of some kind and produces information of some other kind, in short, the proper metaphor for describing the mind is a device that processes information. Within that information processing paradigm the issue of whether the mind is best thought of as a computational device, as opposed to some other form of information processing device, is still seriously being debated. Within the AI community there is a group of people called connectionists who think that, in fact, there are no symbols in the mind. I am basically going to assume that there is something very useful in regarding the mind’s information processing capacity, as in fact, a computational problem. The basic assumption is that the mind is doing some kind of computation. It is not perhaps the kind of computation that an IBM PC does, but it is still computation of some kind. It is also sort of vaguely related to this
whole business of expert systems, even though my emphasis is going to be rather
different from the rule-based approaches that expert systems have been
popularizing. Finally, as I mentioned, design is a very, very complicated type of
activity, and I am going to talk only about a narrow slice of that activity. Although I
will be talking about a narrow slice of the whole activity of design, I think
nevertheless, that the narrow slice has something to say about the design process as
a whole.

Let me first define what I mean by design for my purposes. The designer's task,
viewed as an information processing problem is the following: The designer is
being called upon to give a complete specification of a set of primitive components
under a set of relations so as to satisfy a set of functions or goals. The designer, in
whatever the domain, starts with a set of either explicitly specified or implicitly
specified primitive objects, and then attempts to specify the actual choice of
primitive objects that was made, the parameters describing the primitive objects,
and the kinds of relationships that exist between primitive objects. The reason why
I am defining design at this general level is because, I want to see if in fact,
computer programming can be thought of in these terms. It is clear that
engineering design can be thought of in this form, and it is clear that architectural
design can be thought of in this form even though different primitives are being
manipulated.

The goals to be satisfied in a given design task could include performance goals
of the object being designed, they could include constraints on the process of
constructing the artifact, manufacturing cost, or it could include constraints on the
actual process of design itself, such as, "do not take too long, do not take eight
years to design the artifact, I want it tomorrow afternoon." So, the functions and
goals could include all of these things, and I claim, that this is a sufficiently formal
definition of design so as to be useful for the present discussion.
Introduction

Most first-generation expert systems have been rule-based with a separate inference engine. The rule-based approach has proven to be both practical and profitable, and resulted in a number of expert systems. However, for handling more complex forms of expert problem solving, there is a need for knowledge representation approaches with a richer set of constructs. These constructs should be helpful in capturing other more structured forms of knowledge and should be such that they help organize both knowledge and problem solving behavior for more focused problem solving.

I have been developing an approach to problem-solving that views a complex body of knowledge as being decomposed into a structure of Specialists engaged in collective, cooperative action. Each specialist does the same kind of problem-solving, but contains different domain expertise. The organization of the specialists depends on the problem-solving type and will reflect the conceptual organization imposed on the domain by a human problem-solver. A problem-solver therefore consists of a well organized collection of specialists each doing the same type of activity, while an Expert system consists of a well organized collection of one or more interacting problem-solvers.

I have identified several distinct types of problem-solving--such as diagnosis, which reasons about how to classify a complex description of reality as a node in a diagnostic hierarchy, consequence finding which reasons about the consequences of contemplated actions on complex systems, and design which reasons about providing values for the attributes of some entity which has constraints placed on it. Clearly these are not the only types. Once these types are well understood, I will be in a situation to be able to categorize which kinds of expert problem-solving I know how to mechanize, something with which current approaches offer little help.
Three Classes of Design

Design in General

In general, design is a highly creative activity involving diverse problem-solving techniques and many kinds of knowledge. Very little is known about creativity, although there has been some work on discovery processes and heuristics. Often the goals for a design are poorly specified, and these goals may be altered during the design by feedback from successes or failures. Clearly, as we do not know many of the components of design in general, and as we poorly understand those components we do know about (for example, planning), a general approach to design is currently out of reach.

However, opinions in the literature agree about many components of design activity. There is an element of refinement. That is, descriptions get refined into less abstract forms. Plans are used in recognizable situations where experience has produced a sequence of design decisions that will usually work. Such plans are the result of past planning and validation by repeated use. Design activity often has a rough design phase followed by design proper. Design activity is organized in ways that reflect the structure or functionality of the entity being designed. Similarly the representation of the design is also structured. For example, blueprints will have areas reserved for different subcomponents, or functionally related entities. During the design various restrictions on what is allowable for this kind of entity will be checked at appropriate points, and the initial conditions (ie. requirements) form a starting set of restrictions imposed on the design from outside.

After discussion with practitioners, keeping the above opinions in mind, I have roughly classified design activity into three classes -- although it is clear that there are subclassifications that make the classes overlap in some ways. They vary from completely open-ended activity (design in general) to the most mundane.
Class 1 Design

The average designer in industry will rarely if ever do class 1 design, as we consider this to lead to a major invention or a completely new product. It will often lead to the formation of a new company, division, or major marketing effort. This then is extremely innovative behavior, and we suspect that very little design activity is in this class. For this class neither the knowledge sources, nor the problem-solving strategies are known in advance.

Class 2 Design

This is closer to routine, but will involve substantial innovation. This will require different types of problem-solvers in cooperation and will certainly include some planning. Class 2 design may arise during routine design when a new requirement is introduced that takes the design away from routine, requiring the use of new computers and techniques. What makes this class 2 and not class 1 is that the knowledge sources can be identified in advance, but the problem-solving strategies, however, cannot.

Class 3 Design

Here a design proceeds by selecting among previously known sets of well-understood design alternatives. At each point in the design the choices may be simple, but overall the task is still too complex for it to be done merely by looking it up in a data-base of designs, as there are just too many possible combinations of initial requirements. The choices at each point may be simple, but that does not imply that the design process itself is simple, or that the component so designed must be simple. I feel that a significant portion of design activity falls into this class.

A Class 3 Product. In a large number of industries, products are tailored to the installation site while retaining the same structure and general properties. For
example, an air-cylinder intended for accurate and reliable backward and forward movement of some component will need to be redesigned for every new customer in order to take into account the particular space into which it must fit or the intended operating temperatures and pressures. This is a design task, but a relatively unrewarding one, as the designer already knows at each stage of the design what the options are, and in which order to select them. Thus, there is strong economic justification in attacking this problem.

**Class 3 Complexity.** The complexity of the class 3 design task is due not only to the variety of combinations of requirements, but also to the numerous components and subcomponents, each of which must be specified to satisfy the initial requirements, their immediate consequences, the consequences of other design decisions, and the constraints of various kinds that a component of this kind will have.

While class 3 design can be complex overall, at each stage the design alternatives are not as open-ended as they might be for class 2 or 1, thus requiring no planning during the design. In addition, all of the design goals and requirements are fully specified, subcomponents and functions already known, and knowledge sources already identified. For other classes of design this need not be the case. Consequently, class 3 design is an excellent place to start in an attempt to fully understand the complete spectrum of design activity.

**Classification as Class 3.** If, during an attempt at class 3 design, all of the design alternatives fail, then it is possible that the designer will switch to class 2 activity. This is most likely to happen if the problem is on the border between classes or if the designer has little experience with this type of component and has not yet fully formed a completely satisfactory class 3 approach. I have no way as yet of knowing whether such a distinct inter-class border exists. It appears that experienced
designers are able to judge whether a project is class 3 or not, the main clue being, of course, whether they have designed that component often before with initial requirements that are judged to be similar.

General Description. It should be clear by now that we consider class 1 and class 2 design to be outside the reach of effective contributions from AI technology at present. Class 3 design however can benefit from other work in knowledge-based systems.

It is my working hypothesis that there is a very specific type of problem solving behavior associated with design activities of the class 3 type. Specifically that a hierarchy of conceptual specialists solve the problem in a distributed manner, top-down, by choosing at each stage of the design from a set of plans, thus refining the design. Specialists can use the expertise of other specialists below them in the hierarchy in ways specified by the plans.

An Approach to Class 3 Design

Introduction

A design problem-solver will consist therefore of a hierarchical collection of design specialists, where the upper levels of the hierarchy are specialists in the more general aspects of the component, while the lower levels deal with more specific subsystems or components. They all access a design data-base possibly mediated by an intelligent data-base assistant. I will first describe the design agents, and then the phases of their interaction.

Design Agents

Specialists. A Specialist is a design agent that will attempt to design a section of the component. The specialists chosen, their responsibilities, and their hierarchical organization will reflect the mechanical designer's underlying conceptual structure
of the problem domain. Exactly what each specialist’s responsibilities are depends on where in the hierarchy it is placed. Higher specialists have more general responsibilities. The top-most specialist is responsible for the whole design. A specialist lower down in the hierarchy will be making detailed decisions. Each specialist has the ability to make design decisions about the part, parts or function, in which it specializes. Those decisions are made in the context of previous design decisions made by other specialists. A specialist can do its piece of design by itself, or can utilize the services of other specialists below it in the hierarchy. I refer to this cooperative design activity of the specialists as Design Refinement.

Each specialist also has some local design knowledge expressed in the form of constraints. These will be used to decide on the suitability of incoming requirements and data, and on the ultimate success of the specialist itself (ie. the constraints capture those major things that must be true of the specialist’s design before it can be considered to be successfully completed). Other constraints, embedded in the specialist’s plans, are used to check the correctness of intermediate design decisions. Still more constraints are present in the design database as general consistency checks.

**Plans.** Each specialist has a collection of plans that may be selected depending on the situation, and it will follow the plan in order to achieve that part of the design for which it is responsible. A Plan consists of sequence of calls to Specialists or Tasks (see below), possibly with interspersed constraints. It represents one method for designing the section of the component represented by the specialist. The specialists below will refine the design independently, tasks produce further values themselves, constraints will check on the integrity of the decisions made, while the whole plan gives the specific sequence in which the agents may be
invoked. Typically as one goes down in the hierarchy, the plans tend to become fewer in number and more straightforward.

As each plan is considered to be the product of past planning, refined by experience, one should not expect many failures to occur. However, as not all combinations of values have been handled before or anticipated it is possible for plan failures to occur due to intra-plan and extra-plan constraint violations.

Steps, Tasks and Constraints. I consider a Step to be a design agent that can make one design decision given the current state of the design and taking into account any constraints. For example, one step would decide on the material for some subcomponent, while another would decide on its thickness. A Task is a design agent which is expressed as a sequence of steps, possibly with interspersed constraints. It is responsible for handling the design of one logically, structurally, or functionally coherent section of the component; for example a seat for a seal, or a hole for a bolt.

A Constraint is an agent that will test for a particular relationship between two or more attributes at some particular stage of the design. Constraints can occur at almost any place in the hierarchy. For example, a constraint might check that a hole for a bolt is not too small to be machinable, given the material being used. Constraints will be discussed further when I address failure handling.

The Four Phases

Requirements. The design activity can be considered to fall into four phases. Initially, the requirements are collected from the user and are verified both individually and collectively. For example, it may be reasonable to ask for a component to be made of lead, and for it to weigh less than 5 ounces, but the combination will often be unreasonable. Once it has been established that the system is capable of working with those requirements, a rough-design is attempted.
Rough-design. Rough-design is poorly understood at present, but it serves at least two purposes. First, those values on which much of the rest of the design depends will be decided and checked. If they cannot be achieved then there is little point going on with the rest of the design. This also has the effect of pruning the design search space, as once the overall characteristics of the design are established it reduces the number of choices of how to proceed with the rest of the design. Second, as any mutual dependent attributes can prevent a design from progressing (ie. A depends on B, and B depends on A), rough-design can, as human designers do, pick a value for one of the attributes and use that as if the dependencies did not exist.

It appears at present that rough-design and design share the same conceptual hierarchical structure. However, that remains to be confirmed. The rough-design hierarchy is in general much shallower than the design hierarchy as more general decisions are being made. I am proposing that specialists have both design and rough-design plans to select from depending on the current phase. Not all specialists will need both. It is entirely feasible that phases could be intermixed during problem-solving, but I have chosen to restrict the rough phase to be first, followed by the design phase.

Design. Once a rough-design has been completed satisfactorily, the design phase can proceed. Design starts with the topmost specialist and works down to the lowest levels of the hierarchy. A specialist S begins by receiving a design request from its parent specialist, which might include some design requirements (constraints). It refers to the specification data-base and obtains a list of specification data relevant to its further work. A plan is selected using these data and the current state of the design. For example, if one of the requirements is low cost, a plan with that quality can be selected. The exact nature of the plan selection
process is a matter for further research, and, with a language for plans, will be a major part of the theory of design.

Thus, S fills in some of the design, then calls its successors in a given order with requests for refinement of the design of a substructure. If some of the substructures are independent of each other, then they may be invoked in parallel. The knowledge in the specialist prioritizes the plans, and invokes alternative plans in case of failure by one of the successors. Parts of a plan may indicate immediately that constraints cannot be satisfied. This is considered as failure. When all of a specialist’s plans fail, or when failure can be deduced immediately, the specialist communicates that to its parent.

**Redesign.** If any failures occur during the design process then a redesign phase is entered. If the phase succeeds then a return can be made to the design phase. At the lowest level, failures occur when a constraint fails in some step. The system attempts to handle all failure at the point-of-failure before admitting defeat and passing failure information up to its parent. A step, for example, may be able to examine the failure and then produce another value, in order to satisfy the failing constraint, while still retaining local integrity. This failure handling activity and the associated redesign phase will be discussed later.

Other work on redesign in the literature has concentrated on “functional redesign”, that is, “the task of altering the design of an existing, well understood circuit, in order to meet a desired change to its functional specifications”. Here I use redesign to mean an attempt by a design agent to change a value to both satisfy a constraint while keeping as much as possible intact of the previous design.

**Communication**

The main means of communication in the system is by passing information and control between specialists across the connections forming the hierarchy. In this
way the flow of control is restrained and the system exhibits clear, well-focused problem-solving activity. It remains to be shown whether this form of control is sufficient, but it is based on a belief that Class 3 design systems are "nearly decomposable" and that "the interactions between subsystems are weak, but not negligible". I believe that for Class 3 design the structure is dominantly hierarchical and that interactions are handled by specific strategies.

Information is passed in the form of messages that can, for example, request action, report failure, ask for assistance, and make suggestions. This rich variety of messages is the key to handling subsystem interactions. In addition, one part of the emerging theory of design problem-solving will be the form and content of these design oriented messages.

Other Agents

In general, a collection of design specialists will not be sufficient for the design task, and will, at least, need an intelligent data-base to keep track of the ongoing state of the design. Other specialists outside the design specialist hierarchy could provide calculations, such as stress analysis, and other data-base functions such as catalogue lookup. In a more general design system, requests could be made to other types of problem-solvers. It is perfectly acceptable to consider the human user as one of the problem-solvers, as requests for assistance will occur at well defined points in the design with precise pieces of design to do. The expert system can subsequently check the acceptability of the results provided by using constraints. Here the usual image of the designer controlling the invocation of analysis packages and problem solvers is reversed.

An Instance of Class 3 Design
The Air-Cylinder

Let us consider a real, but not overly complex example for illustrative purposes. In my collaboration with AccuRay Corporation, I have selected an Air-cylinder (AC) as a suitable object for my continuing studies of design problem-solving. My preliminary work on design problem-solving was reported in Brown (1983). I am now working on extending the theory and examining the issues and problems using the AC as a test case. The AIR-CYL design problem-solving system is currently under development using Rutgers ELISP and FRL on the OSU CIS department's DEC system-20 (see figure 1).

The AC has about 15 parts, almost all of which are manufactured by the company according to their own designs, as their requirements are such that the components cannot be purchased. The AC is redesigned and changed slightly for applications with markedly different requirements, and, consequently, the domain is Class 3 in type. Movement is limited by a bumper. The spring returns the piston, and the attached "load", to its original position when the air pressure drops.

Conceptual Structure

An Air Cylinder Designer was interviewed over a period of time, the protocols were analyzed and the "trace" of the design process was obtained. Figure 2 shows the progress of the design over time (from left to right) and the groupings of the decisions (from top to bottom).

The trace was subsequently analyzed to establish the underlying conceptual structure. For example, the Head was clearly treated as a separate conceptual entity, as it occupied a substantial portion of the designers time and effort. The Spring was actually designed by a different person as an essentially parallel activity,
while the rest of the design was "lumped together" by the designer as the third major activity. The fact that the specialists can be fairly easily identified, and that the plans for each specialist are also identifiable and small in number strongly confirms that this is a class 3 activity. On examination it was apparent that this
organization tends to localize dependencies, and allows for parallel design activity—something of which most designers are not able to take advantage.

Concept specific knowledge is located within each of the specialists and coordination of various activities is achieved by each of the specialists. The reason why design teams exist in the real world is for exactly the same computational reason why design teams exist within one mind. The claim is that specialists exist inside the designers head, so to speak. The fact is that there are different concepts and the concepts help the designer organize his or her knowledge concerning the completion of the design task.

Wilcox: Do the members of the design team themselves have minds that contain specialists and do those specialists in turn have minds that contain other specialists?

Chandra: That is a good question, and in fact it used to be called the homunculus fallacy. In the late 18th century, the operations of the mind were often conceptualized in terms of the activities of a homunculus. The theories that conceptualized a homunculus got lost in the same logical problem that you are now raising. The problem is, namely, that nothing gets solved; it is more like passing the buck than solving the problem. The real computational breakthrough involves the fact that not all minds are equal, there are big minds and small minds. The totality of the mind like activities that the mind can do, gets smaller and smaller so that the scope and complexities of the activities are also getting smaller and smaller. By the time a single choice is being made, it is an extremely small mind that performs the activity. Each choice then contributes to more and more complex minds. So the homunculus fallacy is not present in the modern versions of the totality of mind theories.
Wilcox: If one begins with a big mind that has to be broken down into a lot of little minds in order to write an AI program, when the same formula is used as an explanation for the workings of the human mind, it becomes apparent that the homunculus fallacy still remains.

Chandra: If the original mind and the little mind are the same in their degree of complexity, then it does not eliminate the problem.

Wilcox: Even if the minds are not the same, it fails to eliminate the problem because there are an infinite number of possible ways to break the big mind into a lot of little minds. The result will be a multitude of different reductions that cannot be independently verified.

Chandra: You are talking about the problem of doing science in general. Obviously, there are many methods for doing it and only some of them are right. The question remains, how do we tell which one is right? That is, of course, a legitimate question, but another question still remains, which is, how do we stop killing each other over this disagreement and start to do something constructive?

Wilcox: My answer is that the effectiveness of the program will determine which of the various AI approaches is best, but effective program or not, the issue of understanding the working of the human mind is something entirely different.

Chandra: You are certainly welcome to the position, and I may not be able to convince you that you are wrong, but there are reasons to be careful about it.

Sears: There is a practical example of the differences that Steve Wilcox is trying to point out. There are several chess playing programs developed at Northwestern
that began winning left and right as soon as they stopped trying to mimic human chess playing behavior.

Chandra: You as a designer, might in fact, have a different composition than I, that is perfectly acceptable. However, both of us still use the same language of decomposition, the same language of combination, and the same language of internal plan selection, and that is what makes you and me, and Einstein intelligent units, without claiming that you and I have the same set of facts in our heads. In other words, the similarity is not being sought at the level of the exact contents in our heads. The similarity is being sought at the level of the same kinds of operations being present. I am saying that all that is necessary for this idea to be viable is the notion that people, in fact, have domain knowledge that enables them to decompose problems into smaller problems, and that for each problem, there is the potential for the individual to have the knowledge in the form of prototypical ways of solving the problem. My aim, first of all, is not to describe how a particular system works, but rather to say something about what it means to be knowledge based with respect to design. Secondly, to describe the various kinds of representation and generic structures that may play a role in the design process and thirdly, how the kinds of languages that I have been talking about may in fact be useful in encoding some of the knowledge and helping a designer take care of some of the more routine aspects of design. The important thing about these kinds of ideas, is that the computer is being used as a knowledge based problem solver rather than as a calculator or formula executer. There is a serious sense in which an attempt is being made to capture the idea of a mind in a machine, as opposed to merely calculating or crunching formuli.

Zaff: Can you describe how the knowledge is represented within your system.
Chandra: The key to this representation is the idea of a prime hierarchy and plans. What the language accomplishes is the ability to specify a hierarchy of cooperating design specialists. For each specialist, the language makes it possible to enter design plans, plan sponsors and plan selectors. Typically, for each task there are a number of alternative plans, each with a potential sponsor. The sponsor, in essence, specifies the situations in which the plan might be considered appropriate and then applies a weighting factor. The selection, then chooses, on the basis of the supplied information, the particular plan that would be most likely to succeed at a given subtask.

Mitchell: I have a question about the generality of the process of decomposition. It is obviously a central issue in the functioning of an intelligent design machine, but there seems to be some problems associated with it. If we remain confined to the world of discrete symbols, then clearly it becomes possible to break a set of these symbols down into subsets and get various hierarchies. In short, there is a way of looking at things that makes sense in terms of a computer program. However, something happens to this process of decomposition when it is moved into the realm of spatial relations; the notion of a discrete decomposition breaks down. For example, the line drawing in figure 3, can be decomposed as two overlapping rectangles, as a set of five squares, as a Greek cross, or with equal validity as an array of parallel line segments. Each one of these decompositions gives rise to a different world of design.

Chandra: Perhaps I did not make myself clear. The claim is not that every object in the world can be decomposed, or that there exists a unique decomposition. The claim, I am making is that even if a person is a visual designer, he or she still breaks the problem into subproblems, which is not a decomposition in a spatial sense. The
notion of decomposition abstractly speaking, merely says, that the person is going to take the original problem which is too large for them, and attempt to break it down into smaller problems. In some cases, there may be a one to one correspondence, but in other cases there may be no such notion at all. I do not mean to imply that a particular hierarchy actually exists in the world. The world may not be hierarchical, but decomposition is one of the mental strategies that we have for handling the complexity of the problem. The second point is that there are extremely important design problems where the issue is not one of composition, but precisely, one of discovering the emergent form. This is a completely different sort of design task. Typically one might say, "there is a vacant lot in Manhattan,
make me a building for 10 million dollars, " and in such a case there is tremendous amount of work involved in exploring the possible spatial forms that come together. Once it is done, and from then on, it is possible to start decomposing the problem. The practical problem for which architects get paid is one of discovering the visual forms that are possible. This is not a decomposition problem at all, but an exploration problem.

Mitchell: I think what architects do and what a lot of people working graphically do certainly involves decomposition. However, what I think happens with architecture is a form of multiple decomposition; cutting apart the problem in many different ways.

Chandra: That is why I was careful to explain that the work that I have done concretely, is only appropriate to routine design, and the kind of design you are talking about is by no means routine. What you are describing is really a one shot design process, and basically no similar knowledge space has been created anywhere else. For this type of task an enormous amount of effort is devoted to exploration as opposed to decomposition.

Zahner: When a designer is designing even a simple artifact, it seems to me, that the designer will frequently step back from the specific, and open up the world of possibilities only to return to the specific once again. I contend, that any design task, whether it is routine or novel, involves the cyclical process of exploring the world of possibilities and returning to the specific, and when the designer has explored the world of possibilities he or she returns to look at the specific from a different point of view.

Chandra: I did not want to give the impression that I was proposing a complete and exhaustive design model. My question was, "how can the computation aspects
of design be done at all? The answer I have suggested is that there are many types of knowledge structures that are involved in converting the computationally impractical problem into a practical problem. I have stressed the point repeatedly, that design is a very complicated activity and there are many aspects of that activity that I am not prepared to address. I have been focusing on a style of analysis in which knowledge structures are viewed as a way of providing leverage against computational complexity. That is the message I was giving, and I am not claiming that it is an exhaustive program.

Krippendorf: I was wondering, when you said that design is essentially a search for solutions in a knowledge space, where does this space come from? When we are dealing with 3-dimensional things, it seems to be very clear, that the object itself defines the space. However, for real design problems there is a very different kind of space. Your model seems to presuppose that the space is already there, but I think that much of design activity is creating that space within which the solution is found.

Chandra: That is a good point. I think the formally given space is different from the actual space within which the mind operates. Consider circuit design for example, in which the designer is basically dealing with transitors. I am claiming that the problems as originally given is in fact, producing a space which is fairly useless for the mind. The task of the mind is to convert that original problem and the original search space into another search space which enables it to make sense of the original problem. The implication is, not that the mind is searching in the original space, for if it does it is likely to encounter serious problems, rather, it is assumed that the problem induces a logical space which can then be reduced and efficiently searched.
Novak: There is something that is troubling me in all this, and it goes back to the types of knowledge that are brought to a problem. It seems to me that there is one thing missing. It can be described as sort of a Zen relationship, in so far as the problem and solution stand in a relationship to each other, as figures stands to ground. So, one begins by having knowledge about the problem, architects call it typology, and in your model it is called class 3 design. From this perspective it is easy to establish solutions and plans, but what becomes a little more difficult to see, is that there is also a parallel, almost invisible knowledge that is brought to bare on the problem. For example, when we, as designers sit down to solve a problem, we know that we are going to begin first to expand the problem and then later to refine it. It seems to me that the class 2 design problem in your hierarchy involves not so much the knowing of plans, but the knowing of problems; the knowing of the ground rather than the figure.

Chandra: I think you should elaborate a little more on what you mean by problems.

Novak: I mean that if you give me an unknown problem, if you tell me to design a spaceship of which I know nothing about, I still know about people. I still know about physics, and I still know how to look at problems that I do not know about. I have a knowledge of how to go about solving problems for which I have no specific knowledge.

Chandra: My model in fact, includes that capability as the operation to abstract partially indexed functional goal based solutions. Given spaceships for example, there is a method for trying to decompose a problem by identifying various subgoals.
Novak: In a situation where I do not know the various subgoals that comprise the solution to a given problem, I still have a set of ways for finding the subgoals. There is an intermediate step where I do not know the particular knowledge, but I know how I might begin to look around for it, and there is a structure to that intermediate step.

Chandra: Maybe I did not make myself clear. The claim is not that I know how to make a computer do design. Rather, the claim is that to the extent that it is possible to think of the mind as an information processing device, although not everyone accepts the premise, it may be possible to implement knowledge structures in the design process. I am merely claiming that one of the strongest forms of knowledge is the indexed partial plan or partial solution, and that once that knowledge is represented for a given domain, it becomes possible to automate routine design activities within that domain.

Mitchell: Some of the indexing that is needed is actually in the language; spaceship, for example, is like the horseless carriage.

Nadin: Yes, but the indexing is provided in the explanation that Chandra is giving.

Mitchell: I am saying that sometimes the indexing is in the words that are being used.

Chandra: I am not merely claiming that design is the process of using indexed design solutions to solve a problem. We are still left with the problem of having to find the solutions in the first place. There is still a need for a concrete theory or a plan, and that is a non-trivial thing to do. The actual problem of doing design can be thought of itself, as designing a methodology of design which has its own set of plans. This is consistent with my general notion of plans which includes plans for
constructing plans. I believe it is useful in any domain to start from the bottom and start by explicitly representing community shared design knowledge in the form of goal indexed plans. The more this is accomplished, the more the designer can move away from repetitive problem solving, leaving that for the computer and raising the level of the kinds of problem solving that he or she has to do. I am talking about building a community knowledge base for design that can be written in the language I have described.

Waldron: With respect to the topic of decomposition, are you ignoring synthesis, or is decomposition a reversible process.

Chandra: I am not ignoring synthesis at all. I have described synthesis as a process of decomposition and assembly.

Waldron: So you are saying it is a reversible process, but what many designers are saying in their experience of design is that synthesis is not necessarily a reversible process because they expand and change and then come back to look at the problem from a different perspective.

Chandra: It is very hard to follow the method by which a problem is made practical. It is very complex, but it is not mysterious. While performing the decomposition of a problem, the designer is being very skeptical of the decomposition process, always evaluating the process in parallel.

The point that Carol Zahner made is that designers do not simply decompose the problem and immediately solve all of the resulting subproblems. Decomposition is constantly subject to criticism and review. One of the ways to evaluate the decomposition process is to step back from the whole process and see if it makes any sense at all in the context of the problem. This is a more elaborate and realistic
description of decomposition, and is consistent with the general framework that I have defined.

Novak: In terms of expanding and contracting in the search space, one thing that is claimed, is that sometimes in the process of designing something the designer switches domains. For example, if the designer is working with candles, he or she may switch to electricity and invent the light bulb. How do you deal with that in your model, how do you allow that kind of thing to happen?

Chandra: That is a discovery process, and discovery involves serendipity. Basically, the designer thinks he or she is solving problem A when suddenly an idea is discovered relating to the design that was thought to be finished two days ago. So, all that means is that the person may think in their mind that there is only one goal active, while in reality many other goals are simultaneously active. The mind is not a single goal affair, there are multiple goals, waiting to be satisfied, and if something comes along that will satisfy the end state of one of those goals, the executive will be notified to abandon the present goal and switch to the new one.

Wilcox: I want to talk about a fundamental problem that has arisen in this discussion which is the notion that the mind is a computing device. I have frequently heard a statement from information processing theorists that goes something like this, “if you accept this notion of mind as computing device, you may not, but if you do then the rest follows.” Well, I do not accept that notion, and I think there are a lot of people who do not accept it. An adherence to the conceptualization of the mind as a computational device is not without consequences, and ought not, therefore, to be taken lightly. As a result of this information processing metaphor, any computational model is an artifact of the particular primitives that one chooses. You start with the notion that the person is
doing something, in this case you assume that the designer, like you, the computer scientist, begins with a set of primitives and operates upon those primitives. That is the axiom that you start with, and if one rejects that axiom, then that whole computational model becomes fundamentally wrong as a description of human behavior. Not wrong for you as a computer scientist, if you want to create intelligence, but fundamentally wrong as a model of human behavior.

Chandra: I am sure that at some level you are absolutely right, but you need to give me an alternative. Merely saying that the information processing paradigm is not the way it works, is not enough.

Wilcox: There are alternatives and the one that I find most intriguing is what has been called the ecological approach which has arisen out of the work of J.J. Gibson (An Ecological Approach to Visual Perception, 1979 and The Senses Considered as Perceptual Systems, 1966).

Chandra: I am familiar with the Gibsonian position and I do not want to dismiss it. In fact within AI there is a Gibsonian type position which is emerging, called connectionism. As it turns out, there is a real problem with the connectionist position in terms of what such a position is actually able to accomplish. What the connectionist have been able to do is extremely small compared to the more firmly established alternative. That does not necessarily mean that connectionism is wrong, but it still remains to be seen what it will accomplish.

Wilcox: You are talking about AI and I am talking about human intelligence. I am not disagreeing with you that your view is right for computers, I am simply saying that it is not right for human beings.
Chandra: I am not talking about AI, in fact, I want to drop the word A from AI. I want to talk about computational intelligence. I could be partially wrong, but I am obviously not completely wrong, and that is the point I am trying to make. Until an alternative comes along that proves to be better, I will continue believing my position is right.

Wilcox: You asked me for an alternative, now you are not letting me explain what that alternative is like.

Chandra: The point I am making is that the alternative is not working. The alternative still needs to be encoded in a form so that one can say, "here is a machine that does design according to the rules and premises of that approach."

Zaff: Chandra, it seems to me, that the criterion that you have established for a successful theory of the mind is already biased by the information processing viewpoint. I have no problems regarding your conceptualization of intelligence in so far as it explains the behavior of computing machines. What I question is the applicability of such a conceptualization for the human mind. The criterion for a successful theory of human behavior should be whether or not it adequately and succinctly describes human behavior, and not whether it can be implemented in a computing device in order to make such a device behave like a human being.

Wilcox: This issue is something that I have been working on for a while, but certainly, I am not alone. There is a minority within psychology that basically rejects the information processing metaphor and offers in its place an alternative approach. This alternative approach and how it differs from the information processing approach can perhaps best be understood in terms of a simple diagram (see figure 4). The environment is basically described in terms of classical physics.
and our experiences are described in a very different sort of language. Because the two descriptions are inconsistent with each other, it gives rise to the notion that the mind must be the sort of thing that computes a description written in the language of our experiences from an input written in the language of classical physics.

![Diagram of Information Processing Approach](image1)

**Information Processing Approach**

![Diagram of Ecological Approach](image2)

**Ecological Approach**

*Figure 4.* Two Approaches to Human Perception and Action
Some people began to closely examine the situation and realized that the scientist is using the one language to describe the world of inputs and another language to describe the world as it appears to the observer, and that the notion of the mind as a computational device is simply an artifact of taking two different viewpoints of the flow of information as it moves through the system. The alternative involves attempting to describe both the input and output in the same language. The world is the world that is described by the person, and that suggests a new program. Under this alternative, the scientist's problem is to attempt to define variables in the world that are in fact, commensurate with experience, rather than trying to force experience into the classical terms of physics and then develop computational models that explains how the one language gets converted into the other.

One example, comes from studies of facial recognition where the question is, "how do I know how old another person is by simply looking at their face? The traditional view is that the age of the face is in my mind. The only things available in the environment are certain parameters that I compute in order to come up with the notion of that other person's age. Alternatively, if one begins with the assumption that there must be some physical description that would allow the perceiver to determine the age of another person, it sends the scientist on a search for the variable, a search that involves redefining the stimulus, redefining the environmental variables in such a way that it becomes possible to find predictive models that allows one to move from the environmental description to the description of the person's experience. It turns out that using a topological model instead of the euclidean geometric model, allows one to identify the optically specified, and hence, perceivable information that is lawfully related to the apparent age of a person. Cardioidal strain is a transformation that takes place over time as the result of gravity, and a perceiver can simply detect the degree of strain...
in another person's face. The optical information specifying the cardioidal strain becomes the primitive that is perceived, and age is seen as a function of cardioidal strain without any need for a whole series of calculations to enter into the process. That is, if the two descriptions mesh, if the environment is successfully characterized, there is no need for computations and there is no need for a computational model. It is a fundamentally different way of looking at things and the argument is, that this way of looking at things is ultimately going to get us considerably further in understanding human beings. In terms of building artificial intelligence it may be the other way around.

Chandra: No, I disagree.

Nadin: This is a discussion that has been in the artificial intelligence community and outside it for a very long time. If we could solve this, we would get all the NSF grants for the future, but since we have run out of time, this is not to be the case. I want to thank everyone involved in organizing the symposium, and everyone attending it.
APPENDIX

Cognitive Condition of Design Symposium Participants
COGNITIVE CONDITION OF DESIGN SYMPOSIUM PARTICIPANTS

Ruth Branham
Department of Design
University of Kansas
300 A & D Building
Lawrence, KS 66045

Richard Branham
Department of Design
300 A & D Building
Lawrence, KS 66045

Todd Cavalier
513 Emerson Street
Pittsburgh, PA 15206

B. Chandrasekaran
The Ohio State University
Comp. Info SC
2036 Neil Avenue Mall
Columbus, Ohio 43210

Shenchang E. Chen
205 Utica Street
Ithaca, NY 14850

Lisa C. Cherian
163 A. Chittenden Avenue
Columbus, Ohio 43201

John C. Davies
2656 Wexford Road
Columbus, Ohio 43221

Clive Dilnot
Visual & Environmental Studies
24 Quincy Street
Harvard University
Cambridge, MA 02138

John F. Emery
Vic Design Studios, Inc.
830 Xenia Avenue
Dayton, Ohio 45387

Linda Ettinger
University of Oregon
2375 Bailey Hill Road
Eugene, OR 97405
Susan Evans  
2 College Evans  
Providence, RI 02903  

Cameille Goihl  
120 13th Avenue East  
Columbus, Ohio 43201  

Richard K. Gorski  
1502 Garfield  
Marquette, MI 49855  

Raymonde Guindon  
MCC Software Technology Program  
9430 Research Blvd.  
Austin, TX 78759  

Ming-Chylian Ho  
The University of Kansas  
Department of Design  
Lawrence, KS 66045  

Wendy Allyn Hollinger  
9495 High Free Pike  
West Jefferson, OH 43162  

Lorraine Justice  
5759 Middlefield Drive  
Columbus, Ohio 43220  

Judith Killen  
Battelle Memorial Institute  
505 King Avenue  
Columbus, Ohio 43201  

David Kirwin  
Central Ohio Technical College  
University Drive  
Newark, OH 43055  

Klaus Krippendorf, Professor  
University of Pennsylvania  
3620 Walnut Street  
Philadelphia, PA 19104  

Charles Leinbach  
RichardsonSmith, Inc.  
10350 Olentangy River Road  
Box 360  
Worthington, OH 43085
George F. McCleary, Jr.
University of Kansas
219 Lindley Hall
Lawrence, KS 66045-2121

Ronald MacNeil, Principal Research Associate
MIT
Visual Language Workshop Media Lab
20 Ames Street
Cambridge, MA 02139

Peter Maegert
Visual Syntax Design
49 E. Oakland Avenue
Columbus, OH 43201

Richard A. Miller
515 Riverview Drive #101
Columbus, OH 43202

William Mitchell
Harvard University
Graduate School of Design
48 Quincy Street
Cambridge, MA 02138

Kevin E. Mullet
The Ohio State University
380 Hopkins Hall
128 N. Oval Mall
Columbus, OH 43210

Mihai Nadin
The Ohio State University
Art & Design Technology
Cranston Center
1501 Neil Avenue
Columbus, OH 43201

Jack L. Nasar
The Ohio State University
289 Brown Hall
190 W. 17th Avenue
Columbus, OH 43210

Marcos Novak
The Ohio State University
Industrial Design Department
128 N. Oval Mall
Columbus, OH 43210
James O'Grady
O'Grady & Associates
2283 Nottingham Road
Columbus, OH 43221

Earl Powell, Director
Design Management Institute
777 Boylston Street
Boston, MA 02116

John J. Rheinfrank, III
RichardsonSmith, Inc.
10350 Olentangy River Road
Worthington, OH 43085

Duane Rice
The Ohio State University
380 Hopkins Hall
128 N. Oval Mall
Columbus, OH 43210

Laura Rolfe
The Ohio State University
380 Hopkins Hall
128 N. Oval Mall
Columbus, OH 43210

Elizabeth Sanders
RichardsonSmith, Inc.
10350 Olentangy River Road
P.O. Box 360
Worthington, OH 43085

Ronald J. Sears, President
Design Consortium
500 Park Overlook
Worthington, OH 43085

George N. Sepetys
The Comark Group, Inc.
200 Renaissance Center
Suite 2080
Detroit, MI 48243

David K. Snediker
Battelle Memorial Institute
505 King Avenue
Columbus, OH 43201
David Svet
The Ohio State University
Industrial Design
128 N. Oval Mall
Columbus, OH 43210

Whei-Tsu Tahn
205 Utica Street
Ithaca, NY 14850

Dave van Bakergem
Washington University
School of Architecture
Campus Box 1079
St. Louis, MO 63130

Hans & Susan van Dijk
45 W. 27 Street
New York, NY 10001

Manjula B. Waldron, Assoc. Professor
Engineering Graphics
215 Hitchcock Hall
2070 Neil Avenue
Columbus, OH 43210

Patrick Whitney, Chairman
IIT
Crown Hall
3360 South State Street
Chicago, IL 60616

Stephen B. Wilcox, Ph.D.
Director
Design Research Center
3601 Locust Walk
Philadelphia, PA 19104

Brian S. Zaff
The Ohio State University
Department of Psychology
404 C West 17th Avenue
Columbus, OH 43210

Carol Zahner
Battelle Memorial Institute
505 King Avenue
Room 11-5135
Columbus, OH 43201