DE-Vol. 68, Design Theory and Methodology – DTM '94
ASME 1994

ISSUES CRITICAL TO THE DEVELOPMENT OF DESIGN HISTORY, DESIGN RATIONALE AND DESIGN INTENT SYSTEMS

David G. Ullman
Department of Mechanical Engineering
Oregon State University
Corvallis, Oregon

ABSTRACT

Design is the evolution of information. This evolution usually begins with an ill-defined need for a product and ends with exact specifications for production, use and retirement or recycling. A number of researchers have hypothesized that capture and reuse of this evolving information has potential for improving the design process and the reuse of design information. Published work has used the terms *design history*, *design rationale*, *design intent* and *corporate memory* to describe the systems that manage the capture, storage and query of the evolving information. The development of these systems depends on attention to 13 outstanding issues identified in this paper.

I. INTRODUCTION

Design is the process of developing information about an object that has not previously existed. For physical products, the design process begins with customers' needs, knowledge about previous products that filled similar or related needs, and knowledge about manufacturing capabilities and constraints. The process ends with product manufacturing specifications. Typically these manufacturing specifications (drawings, bills of materials, engineering change notices and assembly instructions) are the only record of the evolution of the product. This information is at best a snapshot of the final states of the design process archived for future use.

Recently, the emphasis has been on developing product design specifications (a refined statement of need) early in the design process. One currently popular method for developing specifications is the QFD (Quality Function Deployment method) [Sontow 1993]. The information developed using this technique, or others like it, produce a snapshot of the initial state of product evolution. Here, the information captured is about customers' requirements, product specifications and benchmarks (examples of competitive products). In contrast to the final state documentation, focused on form and structure, this initial state information focuses on the function of the device.

Other information possibly developed and archived is the product development plan. This record of the tasks that make up the design process is usually a bar chart of planned tasks superimposed with the reality of the schedule actually achieved during the product's development. This information, in a coarse way, shows many states in the product's evolution, beginning with the development of product specifications and ending with manufacturing specifications.

Information about the developing product is generally not archived. This information can be viewed as the result of a series of engineering and business decisions each of which is one step in the transformation of the product. Each decision is based on previously developed information, assumptions and guesses. Each decision has an impact on the final product and the manufacturing specifications.

Currently there is interest in capturing information during product development and recording the rationale behind the decisions affecting the evolution of the information. This effort, to capture and manage the intent of the designers in a form that can be queried, opens many questions about modeling and controlling design information. This paper itemizes issues key to the development of systems that manage design information evolution.

II. THE DEFINITION OF DESIGN HISTORY, RATIONALE AND INTENT

Many research communities have studied the idea of capturing and managing the information developed during the design process leading to inconsistent terminology.

A number of authors have explored the concept of recording a <u>design history</u>. Nearly ten years ago Mostow stated that there was a growing consensus in the artificial intelligence community that "An idealized design history is a useful abstraction of the design process" [Mostow 85]. In the late 1980s, this author and his colleagues built an object oriented database that organized information about the design of a simple mechanical system [Ullman 91a, Chen 90, 91]. This database could be queried about the evolution of constraints and the effect of decisions on the artifacts being developed. Baxter, in his thesis, "Transformational Maintenance by Reuse of Design Histories" [Baxter 90] used the concept of a history to aid in the redesign of computer code. All three of these authors saw the concept of a design history as including a record of the rationale behind design decisions and of the intent of the designers.

Since Mostow, the artificial intelligence community has been active in developing design rationale systems. A workshop held in 1992 [Lee 92] defined the term "rationale" as "an explanation that answers a question about why an artifact is designed as it is." Baxter, in his thesis, refines the definition, "Design rationale: An information structure that justifies how the implementation (consequences of the design selections) satisfies its specification." This definition emphasizes the structure of the design information and tracking the relation of decisions back to the specifications.

A third community, that organized around development of the STEP IV standard, uses the term design intent. Their use of design intent refers to the cause and effect relationships among product data. In an unpublished document, a STEP IV researcher states "Generally, the term intent means the purpose or plan for performing activities. During product design these activities transform a set of requirements to the final specifications for production. In a basic sense, the intent is the blueprint for the evolution of the requirements into the production specifications. This blueprint not only has information about the development of the geometry, but also on the evolution of the product function and behavior, the rationale underlying design decisions, and the influence of business activities."

In some literature, the term corporate memory is used. This term emphasizes the feeing that the information managed is beyond that associated with the traditional artifact as it includes business information as well. In the results of the first CERC Workshop on Enabling Technologies [Nichols 92], the discussion on corporate memory was in terms of design histories and rationales.

As can be seen, there is little difference between the terms design history, design rationale, design intent and corporate memory as defined above. Thus, it is assumed that these terms all refer to systems with the characteristics discussed in the previous paragraphs. Throughout the remainder of this paper, systems with these characteristics are called **design intent systems** and these systems allow query of information that supports **design rationale**.

Issue 1: A consistent definition of systems structured to allow capture and query of product evolution information needs to be developed.

III. THE USE OF DESIGN INTENT SYSTEMS

Mostow [Mostow 85] itemized the following roles for a software design intent system: documentation, understanding, debugging, verification, analysis, explanation, modification and automation. These roles support the basic goal of design intent systems - *to organize information needed to answer questions about the evolution of the designed artifact and the process through which it matured*. The users of such information include four main groups of people. They are discussed in their order of involvement which is not necessarily their order of importance.

The first user of design intent information is the original designer querying his/her own work. Typically, designers do not clearly remember why they made earlier design decisions and forget outstanding issues [Ullman 88, Stauffer 88]. For designers, a design intent system could be a glorified design notebook used to document their design progress in a manner that is easily recovered.

The second user of design intent information is another member of the design team insuring (s)he is current and confident of the design progress. A project at National Cash Register, [Yakemovic 89, 90] provided a design team with a tool to record and query the decisions made in the development of software. The entire project, undertaken by five software engineers, addressed 2,260 design issues and is by far the largest study of its type. Some of the results from this study are: 1) The tools provided a shared memory for the design team. The history of decisions made and recorded in the tools was easily reviewed and used by members of the design team and management to understand the state of the project. 2) The use of the tools helped the design team detect design issues that had "fallen through the cracks." This debugging feature resulted in an estimated savings of 3-6 times greater than the cost of using the tools. 3) The use of the tools helped the team to understand more quickly the problem they were trying to solve and 4) The tools aided in making team meetings more productive by structuring the information and by establishing and focusing the agenda.

The third group of users of design intent information are managers desiring to keep abreast of design progress. Yakemovic [Yakemovic 90] found that the use of the tools supported communication with other levels of the organization. People not at a design team meeting could easily discern what was discussed, not just the final decisions.

The forth users of design intent information are re-designers who need to understand the relations of information during their effort. In a small set of experiments [Kuffner 91], the value of design intent information on a redesign situation was tested. One of the conclusions of this study was "mechanical design engineers are interested in design information other than that which is contained in standard design documentation." The study provides much evidence to support this statement as will be presented later in this paper.

IV. DESIGN INTENT SYSTEM INFORMATION

This section provides an introductory list of the types of information which might be included in a design intent system. In the following list, the terms underlined are heavily used in this paper and thus, are briefly defined.

In general, design intent systems should include information concerning:

Problems or <u>issues</u> addressed (e.g. business issues, planning issues, artifact design issues)
<u>Alternatives</u> considered (e.g. tasks, assemblies, components, features, functions, materials, dimensions, tolerances)
<u>Arguments</u> for or against alternatives (e.g. qualitative discussion, quantitative analysis, rules, standards)
Methods used to <u>evaluate</u> alternatives
<u>Constraints</u> and <u>requirements</u> used, relaxed and their sources (constraints are restrictions developed during previous design decisions and requirements are stated design goals)
<u>Assumptions</u> made
<u>Decision history</u> and the <u>rationale</u> for each decision
<u>Product development plans</u> used
Changes in <u>attribute</u> or <u>parameter</u> values (these terms are used interchangeably)

In order to describe systems that can support these types of information the remainder of this paper is organized around five primary characteristics of design intent systems: **granularity, formality, system type, capture mechanics and query needs.** As will be shown, these measures are not independent and each raises issues that must be addressed in the development of design intent systems.

V. GRANULARITY

The information generated during the design process can be aggregated in many different ways. On the finest level, individual designers or teams of designers make micro-decisions at the rate of about one per minute [Ullman 88]. These episodes (the term used by cognitive psychologists) typically concern each step in solving equations, each change in attribute values, what to do next and so forth. One effort to build a design history system that tracked decision making at this level of granularity showed that this fineness was necessary for information capture completeness but, was very difficult to implement and was not realistic for an operational system [Ullman 91a].

A coarser level of granularity is at the task level. Here a <u>task</u> is defined as the work of a single individual on a single, coherent piece of the design effort. The information developed during a task is usually reported to others working on the design project. In developing these results, there may be many hundreds or thousands of micro-decisions (episodes). Thus, design intent systems tied to the task level may miss much important information concerning alternatives considered, evaluations completed and assumptions made. Lakin [Lakin 92] attempted to resolve this problem by allowing designers working on individual tasks to tag unstructured information in a computer based notebook to identify key ideas. This tagging resulted in links between informal text and graphics and only took about 20-30 seconds/hour of recorded information according to Lakin. Additionally, about 20 minutes/week was needed to update the "idea tag table," a table that serves as an index to the tags. Informal testing showed that this method of information tagging may be useful within design tasks.

Many researchers have modeled the design process at the <u>issue</u> level of granularity [Yakemovic 89, Nagy 92, Lee 90]. An "issue" is fairly loosely defined in these studies. In the study of Yakemovic an average issue lasted about two hours.

Coarser information is developed in design projects or programs. Here <u>projects</u> are defined as design activities performed by single discipline teams whereas <u>programs</u> require team members from diverse disciplines. The design effort at project/program level is of the granularity normally seen in corporate product development plans. Thus, the information only represents snapshots of refined work that is ready for communication to other projects or programs. This is the level of information often handled by commercial systems such as IBM's Product Manager 6000™ and SDRC's DMCS™ system.

The enterprise sees the design process as a coarse network of internal programs and external vendors and customers which combine to develop new products. From this corporate level, business and technical decisions are made only about very coarse information. The systems mentioned in the previous paragraph also apply to this level of product design activity.

One long term effort to model systems at varying levels of granularity is IDEF [DEF 81]. The various versions of IDEF break information transformations into activities or operations. The granularity of the transformation can range from an episode to enterprise level tasks. Although IDEF may be a tool for modeling design intent system information, it does little to aid in establishing the level of granularity for the information.

> Issue 2: Study is needed to determine the levels of information granularity necessary for design intent systems. There may be different levels for different activities in the design process.

VI. FORMALITY

Computer techniques to manage formal information are better developed than those for informal information. Unfortunately, much of the information that may be important in a design intent system is informal. Evidence for this assertion comes from a studies of design information evolution. McGinnis [McGinnis 92] studied the evolution of commitments (constraint development) in a single part of a single product. This study concluded that during conceptual design 55% of the information was abstract and difficult to formalize. Even during the detail design stage 17% of the information was informal. Throughout the entire development of the part, fully 34% of the information was abstract and thus difficult to formalize. Lakin's tagging [Lakin 92] of important information provides minimal formalization yet provides some design intent information. Any formality beyond simple tagging will require a model of the information represented. Information models will be discussed in Section VII.

> Issue 3: Research is needed to determine how much of the information managed in a design intent system needs to be to formalized.

Historically, the mechanical engineering and computer science communities have been very successful in formalizing much of the geometric information that characterizes the evolving product. However, even in this refined area, information on tolerances and sketches of abstract information remains important yet difficult to manage [Ullman 90]. Efforts have been made to represent, and thus formalize, function through a set of key words that give a complete set of possible functions [Koller 85] and mathematical representations based on bond graphs [Rinderle 90].

A number of researchers have semi-formalized the design process as will be discussed later. However, on the whole, there is a large amount of information in the design process that can not be adequately represented.

> Issue 4: Methods must be developed to formalize important design information such as that describing product function and the design process. Currently only geometry, parameterized product information and limited planning information is formal.

VII. TYPES OF DESIGN INTENT SYSTEMS

In order to discuss the structure of design intent systems five types are defined. This classification is based on existing literature but, there is no guarantee that the listing is complete. In each of the types of systems the information must be captured, stored and queried. During capture information flows from the designer through a capture interface to the data store. During query, stored information is retrieved by the user through a playback interface. Table 1 shows the activities necessary by the designer, capture interface, data store, playback interface and user for each of the five types of design intent systems. Each type is discussed in detail in the following text.

Type I. Direct History System

The simplest type of design intent system records a chronological history of the design process. One form of this is the common designer's notebook where the designer writes and draws on paper to record the important activities and information concerning the design project. The paper serves as the storage medium. Information is informal and chronological. Any indexing of the notebook is the result of a special effort by the designer or the reader (user). Further, designers do not always record the reasoning behind decisions in the notebooks. This type of information often must be inferred by the user from the designer's activities and results that were recorded.

The comments made about the notebook can be expanded to cover the design records for a team or an entire project. These are, at best, a chronological record of meetings, design revisions and analysis results.

Current technology allows for a computerized designer's notebook. To some crude degree, this is accomplished through a drawing revision history available in most commercial CAD systems. Systems such as Microsoft Notebook™, available on pen based systems also provide this facility.

> Issue 5: Can computer based direct history systems provide some significant fraction of design intent information? Evidence from paper based notebooks imply that they are very limited in their potential.

Type II. Designer Input Rationale System

Making the common design notebook a more useful design intent system requires that the information be more than temporally related. As discussed earlier, Lakin [Lakin 92] has taken the approach of developing a computer based notebook which allows the designer to go back over his/her work and tag the important information. Thus, the designer first does the work in a computer based notebook and then reviews it noting valuable results, drawings, notes and so forth. It is Lakin's assumption that the designer can not explain his design actions while performing design activities. This assumption is commonly accepted based on human cognitive limitations [Miller 56]. This history based approach requires little additional work and can manage informal information, however it lacks completeness and consistency. Additionally, it assumes that the original designer knows the structure of the information needed by the user.

Lakin's effort is only one way of approaching Type II intent systems. In informal experiments by the author, a personal design notebook (paper based) was reviewed periodically during the design process and the issues, alternatives, evaluations and decisions identified and noted in the margins. This served to give retrospective organization to the argumentation used in the design effort greatly clarifying the design process followed, keying the author to new artifact ideas and showing promise for organizing design process activities. The realization that added effort results in added benefits differs little from those of Yakemovic and Conklin at NCR [Yakemovic 89].

TABLE 1: TYPES OF DESIGN INTENT SYSTEMS

Designer(s)	Capture Interface	Data Store	Playback Interface	User
Type I: Direct History System				
Designer writes or draws on paper		Stored in notebooks	User reads notebook and infers rationale	
Works on computer or scans paper documents	Text and drawings are informal	Text strings and drawings or pixel maps	Display of stored data	Infers rationale from stored data
Type II: Designer Input Rationale System				
Makes effort to record reasoning	Accepts rationale with product and process information	Data informal but annotated with rationale information	Display of stored data	Rationale realized by designer is available
Type III: Model of Artifact and/or Process System				
Uses structured interfaces during design	Accepts information through structured interfaces	Data formalized by model	Display of attributes and relations in model	User queries model
Type IV: Parametric System				
Uses parametric tools on computer	Tools structured to capture data about artifact in terms of parameters and their relations	Data is stored as parameters and their relations	Display of parameter values and relations (possibly different from those captured)	User queries system about parameters and relations
Type V: Inference System				
Works in any of the other types of intent systems	Data captured consistent with system type	Data stored consistent with system type	Information inferred from data stored to meet user requirements	User queries about stored and inferred data

Issue 6: It is generally assumed that a successful design intent system must add little or no burden to the designer's effort. However, if use of the system assists the designer while design intent information is captured, then what is a reasonable tradeoff?

Type III. Model of Artifact and/or Process System

Model based design intent systems require that the structure of information about the artifact being designed and/or the process being followed is known and that the data store is structured to receive instantiations representing specific design problems. This type of system is limited to routine design situations or design with known information relationships.

As shown in Figure 1, four different classes of models exist, two each for the product and the design process. The discussion below focuses on each of these non-exclusive classes.

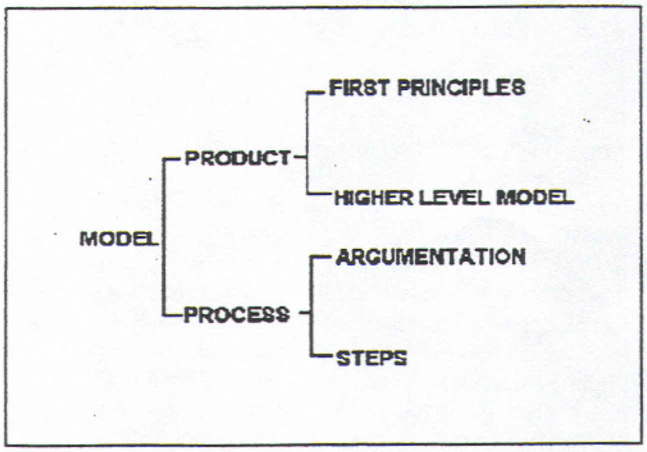

FIGURE 1: MODELS OF DESIGN INTENT SYSTEMS

First Principle Models of Products. In some models of the ideal design process, designers begin with first principles when developing concepts [Pahl 84]. In reality, designers only resort to first principles when doing detailed analysis or when they are having difficulty trying to generate ideas [Fricke 93]. This is not to say that design intent systems should not have an underlying structure based on first principles, rather that such systems are far from being developed.

Higher Level Product Models. A domain specific, refined model of the product being designed is a higher level model. Knowledge based systems are higher level models that have, to some degree, the design intent inherent in them. As an example, PRIDE [Mital 86] is an expert system used daily by designers doing feasibility studies for new copiers at Xerox [Dym 1994]. PRIDE assists in the design of the paper transport system that is central to all copiers. PRIDE simulates rather closely the actual design process that human designers use. It uses a top down process of identifying and satisfying design goals and sub-goals. Depending on the requirements and attributes of the artifact, the goal structure evolves as the design progresses. The authors of PRIDE spent about three work years eliciting information from experienced designers in order to develop the process logic. PRIDE uses heuristic, relational, inferential and algorithmic schemes to process both information about the product and the process. Although PRIDE is not a design intent system, by tracking the evolution of the attribute values and the decisions made by PRIDE's logic gives much of the design rationale for the paper path's development.

More recently Garcia [Garcia 93] developed "an initial design model able to generate and explain standard design decisions." In this system, Garcia uses abstracted knowledge about the design of Heating, Ventilating and Air Conditioning (HVAC) systems. The software developed integrates the process with the product. Information about the designed artifact and the constraints on it are the elements of the decision itself. Garcia simulates the behavior of an apprentice who is observing the designer asking questions whenever not understanding what the designer did.

Argumentation Model of the Process. In an argumentation model the system records the decision logic structure in a manner that allows for easy query. Information is identified by its role in the decision making process.

Most current argumentation systems are based on IBIS. Horst Rittel developed IBIS for organizing the deliberation process that occurs during complex decision making [Rittel 73]. A computer tool to capture design histories and support computer mediated teamwork (groupware), gIBIS (graphical IBIS) [Conklin 88, Yakemovic 90], uses the IBIS model. Groupware refers to "computer-based systems that support two or more users engaged in a common task, and that provide an interface to a shared environment" [Ellis 88].

IBIS type systems model the decision making process in terms of issues (tasks, questions or problems), alternatives (proposals or concepts), arguments (evaluations) and decisions. Issues are oriented toward the design process itself (planning issues) or toward artifacts (specification issues) [Ullman 88]. In most embodiments of IBIS the artifact information in the alternatives and arguments has been left informal. However, researchers at Oregon State University have formalized the modeling of the artifacts [Nagy 92, Chen 90, Chen 91]. This work uses the decision network to index the changing state of the evolving artifacts. Through this system the design model describes the sequencing, composition and dependence between the decisions. It further describes the constraint development and propagation and the dependence on the design specifications. Lee has taken a similar approach in the development of DRL (Decision Representation Language) [Lee 90].

Where IBIS and DRL use natural language text to describe the decision making process, Klein [Klein 93] developed a structured language that attempts to represent the product and the process. This language, DRCS, is quite general and is still in development and untested.

A non-IBIS based argumentation system is the QOC system [McLean 91]. In QOC the model is in terms of: Questions-identifying key design issues, Options - providing possible answers to questions and Criteria - for assessing and comparing options. This model is intended to be used retrospectively, not at the time of actual design effort.

Modeling Design Process Steps. Another approach in modeling the design process for well understood problems is to follow a template of the design process steps. This "cook book" approach is embodied at the project level by very detailed product development processes such as that used by Xerox [Xerox 88]. Popular design methods such as the quality function deployment (QFD), Pugh's method or design for assembly (DFA)) [Ullman 92] outline steps that, if followed, give structure to the rationale behind decisions made. Unfortunately, design methods do not cover all the decisions made during product evolution nor are they consistent in their treatment of information.

There are a number of commercial tools that support the product development process. For example, the goal of IBM's ProductManager^{TM} 6000 and SDRC's Data Management and Control System DMCS^{TM} are to provide an automated approach to: initiate, evaluate, plan, define and document design projects and changes to existing products. In these systems, information (e.g. CAD drawings, bills of materials, Specifications, engineering change orders and management information) is created and placed in a folder then distributed. Although this process disseminates information, it is limited in organizing information needed in a design intent system.

Issue 7: It would seem that models can be used to carry most of the design intent information load in well understood areas. Such areas include routine designs that can be captured in higher level models and well established product design processes that can be captured as design process steps. However, many design efforts are not routine and the design process is often fluid. Methods to model these situations such as argumentation and first principle methods need to be further developed.

Type IV: Parametric System

In parametric and variational systems, constraint relationships are built as the design is developed. In other words, a basic structure of relationships between the variables is instantiated as the artifact evolves. Most parametric systems, such as that marketed by Parametric Technology Inc. are predominantly graphic based systems. Thus, the constraints are between geometric primitives. However, variational systems allow equations that model behavior to also be included in the constraint network.

Parametric and variational systems allow for design reuse. Thus it is easy to find the answer to queries about the affect of parameter (attribute value) changes and sensitivities. In this manner these systems have captured some of the intent behind the information modeled. One limitation of these systems is that they can only be used with geometric decisions and decisions about behavior that can be formally modeled. Additionally, these systems do not model the decision structure. If the value of a parameter (i.e. the length of a part) is queried, parametric systems will give information on this length's dependency on other dimensions of the part. However, it does not give the rationale for the relations, the order in which the dependency was formed nor the arguments for the current value.

Issue 8: Most of the geometric features of an evolving product can be parameterized sufficiently to allow design capture by parametric systems. It is not yet clear how well such systems support design intent needs nor how far parametric/variational systems can be extended beyond geometry and behavior. Additionally, it is not clear how to capture functional, temporal, argumentation and rationale information with such systems.

Type V: Inference System

Some queries can be answered directly based on information captured. For example, the dimension of a component initially entered on a detailed drawing of a part can easily be retrieved. However, the dimension on an assembly composed of a number of parts may not be a stored piece of information. It can, however, be calculated if needed. Gruber [Gruber 93] claims that it is not sufficient just to capture, store and retrieve the same information. He observes: "Rationales (intents) are constructed and inferred from stored information rather than as complete answers." In other words, design intent systems may have to answer questions that require information different than that which was captured. The questions that arise during query may not be answerable with only the information of the original design. This implies that the data must be structured during capture or storage so that answers can be developed to needed questions.

Gruber also claims: "Rationales are not just statements of fact, but explanations about dependencies among facts." Artifact design facts are dependent through the process that developed them. Thus, a design intent system must capture dependency relationships and be able to reason with them.

Issue 9: Study is needed to answer the question "To what degree can captured information satisfy design intent queries?" Assuming that inference is necessary to satisfy the query needs, then the development of design intent inference methods is also necessary.

In actuality, a useful design intent system may be based on one or more of Type I-IV design intent systems and have to infer information not available from them. The amount of inference needed will be dependent on the capabilities of the other types of systems.

Any usable design intent system will probably utilize all five types of design intent modeling. There are certain portions of any design problem that can be parameterized, other portions that will have formalized models, others that can best be annotated and some that can only be recorded as a chronological history. A robust design intent system will allow for all these types to be handled seamlessly and concurrently.

Issue 10: The major issue in developing a design intent system is to determine when information can be managed as direct history, as designer input rationale, as modeled parameters, as automatically parameterized by the system or as requiring inference. It is suspected that a successful design intent system will utilize all of these types.

VIII. CAPTURE MECHANICS

One of the most important and least studied aspects of design intent systems is that of data capture. Garcia [Garcia 92] in the development of her system made the following assumptions about the capture of information:

1. Rationale should be captured in the context of use when a design is made.
2. Retrospection does not work. (Note that this is in conflict with Lakin's approach.)

3. Rationale can be captured by explaining decisions. (This is the basic assumption behind argumentation systems.)
4. New knowledge can be anchored in old knowledge.
5. System and user should cooperate.
6. Knowledge transfer is two way.

These assumptions imply that the designers need to work on a computer based system that supports all design activity. This allows the rationale to be captured during design so the context of the information is clear. Since the system is capturing all the information the designer's are developing, new information can be built on previous information and retrospection is possible when desired. If such a system supports the design process, then the decision net work complete with explanation can be developed by following the design activity. Further, if this system is robust with integrated applications to support design activity, then the user and system will be cooperating in a two way information exchange.

Where Garcia's system honored the above assumptions for HVAC system design, this author [Ullman 91b] fantasized about a generic system called DUDA (Dave Ullman's Design Aid). This hypothetical system would rely on a pen based design notebook that would support Type I-V design intent systems. The basic assumption being that this system is so facile in supporting common design activities, that designers would want to use it. The capture of design information would be, for the most part, transparent to the user as DUDA would be so rich in Type III and Type IV tools. DUDA is still in development.

Issue 11: The capture of design intent information is virtually unstudied. Designers are not trained to express this information and, with the exception of a few experiments, little has been done beyond conjecture. It is clear, however, that capture is key to design intent system success and must not add a burden to the design process.

IX. DESIGN INTENT QUERY NEEDS

A major design intent system development issue is understanding the user's information needs. Gruber and Russell [Gruber 92] synthesized the results of five studies (including those by Garcia, McLean and Kuffner discussed elsewhere in this paper) into one list of generic questions. The exact questions are too detailed for this paper and the classification of them more important. Thus, only an outline of the query classification issues will be covered here.

During a query the questions must refer to information about the design effort to date. Any query that requires new information to be developed is design activity and should itself be captured.

The discussion below is divided into product and design process. This division does not imply that the information about product and process are independent, on the contrary, the answer to many of the questions requires both product and process information.

Product Questions

Product questions generally address an attribute (e.g. parameter) of an object which has a specific role in the product design effort. Thus, questions are in terms of the triad (object, attribute, role).

The "object" is the focus of the question. Typical objects queried are product, assembly, component, interface, feature, environment, or user. Kuffner classified questions and conjectures as assembly (17%), component (33%), interface (16%) and feature (34%). The percentages are the relative occurrence of each type of objects in his data. He counted product data with assembly data and did not consider the environment or the user of the product during his study.

The "attribute" of the object considered can range over name, geometry, dimension, tolerance, material, behavior, function, cost, operational state, or other class. Here "other class" means any classification scheme used during query.

During the design process objects can play many different roles. Query is often directly related to the role. Typical roles are state (versions), proposal, requirement and example. The role "state" refers to the instantiation of the object's attributes at any point in it's recorded history (e.g. working, preliminary, approved, released). The role "proposal" refers to objects or attributes of objects that were proposed but did not become part of the product. The "requirement" role denotes the ideal or target state of the objects. Finally, "example" information is that describing previous cases that were considered during the design of the product.

One further note about questions is that they may be asked relative to any of the operating conditions that the product experiences. Thus, each question can be modified by the clause "in operating condition."

Process Questions

Process questions are divided into questions about the issues faced, the alternatives developed to satisfy the issue, evaluations of the alternatives and decisions made. This process model is based on IBIS and is robust to the point that it can be used to represent work at any level of granularity.

Few questions can be answered with a simple history recording of the design evolution. Some can be answered through annotation or tagging. Most require Type III, IV or V systems and some amount of data formality.

Issue 12: What is a complete set of design intent queries? Or, what is a useful set of design intent queries?

In developing a system that supports query, a concern is whether anybody will use it? Kuffner [Kuffner 91] noted when his subjects sought information from the drawings, the specifications (original or redesign) or the examiner who was filling the role of a design intent system. One subject, who was not trained that the examiner was a good source of information, did not confirm 60% of his questions. In other words, the designer could not support the answer to his questions through the documentation provided (standard drawings and design specifications) nor did he ask the examiner for support. This is not to say that 60% of his answers were wrong, only that he had no direct evidence to support or deny his best guess at the answer. Two other subjects, both trained to know that the examiner was a good source of information about the original design, left only 27% unconfirmed. These designers risked a smaller chance of using wrong information in the redesign effort. This experiment, while lacking in sample size and use of an actual system, is the only known data on the potential of a design intent system.

> Issue 13: The usefulness of a design intent system can be measured in terms of its ability to supply users the correct information when they may have guessed incorrectly without system support. How can the usefulness of these systems be measured and can the potential of such systems be tested without building them?

X. CONCLUSIONS

The concept of a system that captures, manages and allows queries of evolving design information has great potential but has only recently been studied. In this paper an effort has been made to summarize the work to date, organized by the important issues that need to be addressed. Each issue raised begs further study and experimentation.

XI. REFERENCES

[Baxter 90] "Transformational Maintenance by Reuse of Design Histories," Baxter I., Ph.D. Dissertation, Information and Computer Science, University of California at Irvine, 1990.

[Chen 91] "A Computer-Based Design History Tool," Chen, A., Thesis for the Department of Mechanical Engineering, Oregon State University, July 1991.

[Chen 90] "Design History Knowledge Representation and Its Basic Computer Implementation," Chen, A., B. McGinnis and D.G. Ullman, The Proceeding of the Design Theory and Methodology Conference - DTM 90 -, ASME DE - Vol. 27, September 1990, pp 175-184.

[Conklin 88] "gIBIS: A Hypertext Tool for Exploratory Policy Discussion," Conklin, J., and M. Begeman, Proceedings of the Conference on Computer Supported Cooperative Work, September 1988, ACM, pp 140-152.

[Dym 94] *Engineering Design: A Synthesis of Views*, Cambridge University Press, To be published 1994.

[Ellis 88] "The Groupware Project: An Overview," Ellis, C.A., S.J. Gibbs and G.L. Rein, MCC Technical Report Number STP-033-88, January 1988.

[Fricke 93] "Empirical Investigation of Successful Approaches when Dealing with Differently Precised Design Problems," Fricke, G., International Conference on Engineering Design, ICED 93, The Hague, August 1993, pp 359-367.

[Garcia 92] "Acquiring Design Knowledge Through Design Decision Justification," by Garcia, A.C.B., and H.C. Howard, AIEDAM 6(1) 1992, pp 59-71.

[Garcia 93] "Active Design Documents: A New Approach for Supporting Documentation in Preliminary Routine Design," Garcia A.C.B, H.C. Howard and M.J. Stefic, T.R. #82, Center for Integrated Facility Engineering, Stanford University, February 1993.

[Gruber 92] "Derivation and Use of Design Rationale Information as Expressed by Designers," Gruber, T., and D. Russell, Stanford Knowledge Systems Lab., KSL 92-64, 1992.

[Gruber 93] "Generative Design Rationale: Beyond the Record and Replay Paradigm," Gruber, T., and D. Russell, KSL 92-59, updated February 1993.

[IDEF 81] "Integrated Computer-Aided Manufacturing (ICAM): Architecture Part II: Volume IV - Function Modeling Manual ($IDEF_0$)," AFWAL-TR-81-4023, Volume IV, June 1981.

[Klein 93] "Capturing Design Rationale in Concurrent Engineering Teams," Klein, M., IEEE Computer, January 1993, pp 39-47.

[Koller 85] *Konstruktionslehre fur den Maschinenbau*, R. Koller, Springer Verlag, 1985

[Kuffner 91] "The Information Requests of Mechanical Design Engineers," Kuffner, T., and D.G. Ullman, Design Studies, Vol. 12, No. 1, January 1991, pp. 42-50.

[Lakin 92] "Mapping Design Information," Lakin, F., V. Baya, D. Cannon, M. Brereton, L. Leifer L, and G. Toye, AAAI 92 Workshop.

[Lee 90] "SIBYL: A Qualitative Decision Management System," Lee, J., Chapter 5 in *Artificial Intelligence at MIT: Expanding Frontiers*, edited by Winston, P., and S. Shelland, MIT Press, 1990.

[Lee 91a] "What is Design Rationale?", Lee, J., and K. Lai, Human-Computer Integration, 6(3&4), pp 251-280, 1991.

[Lee 91b] "A Comparative Analysis of Design Rationale Representations," Lee, J., and K-Y Lai, Center for Coordination Science TR# 121, MIT, Cambridge MA, May 1991.

[Lee 92] "Summary Report of AAAI '92 Workshop on Design Rationale," Lee, J., Concurrent Engineering Research in Review, Autumn 1992, Vol. 4, Concurrent Engineering Research Center, UNA, 29-32.

[McLean 91] "Questions, Options, and Criteria: Elements of Design Space Analysis," McLean, A., R.M. Young, V.M.E. Bellotti, and T. Moran., Human Computer Interaction, Vol 6 pp 201-250, 1991.

[McGinnis 92] "The Evolution of Commitments in the Design of a Component," McGinnis, B., and D.G. Ullman, Journal of Mechanical Design, Vol. 144, March 1992, pp. 1-7.

[Miller 56] "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity For Processing Information," Psychological Review, Vol 63, No. 2, 1956, pp 81-97.

[Mital 86] "PRIDE: An Expert System for the Design of Paper Handling Systems," Mital, S., C. Dym and M. Morjaria, IEEE Computer, May 1986

[Mostow 85] "Toward Better Models of the Design Process," Mostow J., AI Magazine, 6(1) pp 44-57.

[Nagy 92], "A Data Representation for Collaborative Mechanical Design," Nagy, R.L., and D.G. Ullman, Research in Engineering Design, Vol. 3, No. 4, 1992, pp. 233-242.

[Nichols 92] "CERC's First Workshop on Enabling Technologies for Concurrent Engineering", Nichols, D., et al, Concurrent Engineering Research in Review, Summer 92, Vol 3, Concurrent Engineering Research Center, UNA, 14-15.

[Pahl 84] *Engineering Design*, Pahl, G., and W. Beitz, Springer Verlag, 1984.

[Rinderle 90] "Automated Modeling to Support Design," Rinderle, J., and L. Balasubramamiam, ASME Design Theory and Methodology - DTM90-, DE-Vol.27, Chicago, 1990, pp 281-290.

[Rittel 73] "Dilemmas in a General Theory of Planning," Rittel, H.W.J., and M.M. Webber, Policy Sciences, 4, 1973, pp 155-169.

[Shum 91] "A Cognitive Analysis of Design Rationale Representation," Shum, S., Ph.D. Dissertation, Dept of Psychology, University of New York, December 1991.

[Sontow 93] "Interaction of Quality Function Deployment with Further Methods of Quality Planning," Sontow, K., and D.P. Clausing, LMP-93-005, Laboratory for Manufacturing and Productivity, MIT, April 93.

[Stauffer 87] "An Empirical Study on the Process of Mechanical Design," Stauffer, L., Thesis for the Department of Mechanical Engineering, Oregon State University, September 1987.

[Ullman 88] "A Model of the Mechanical Design Process Based on Empirical Data," Ullman, D.G., T.G. Dietterich, and L. Stauffer, Academic Press, AIEDAM, 2(1), 1988, pp. 33-52.

[Ullman 90] "The Importance of Drawing in the Mechanical Design Process," Ullman, D.G., S. Wood, and D. Craig, Computers and Graphics, Special Issue on Features and Geometric Reasoning, Vol. 14, No. 2, 1990, pp 263-274.

[Ullman 91a] "Design Histories: Archiving the Evolution of Product," Ullman, D.G., Proceedings of the DARPA Workshop on Manufacturing, Salt Lake City, February 1991.

[Ullman 91b] "The Foundations of the Modern Design Environment: An Imaginary Retrospective," Ullman, D.G., Proceedings of the Workshop on Research in Design Thinking, Delft University, May 1991.

[Ullman 92] *The Mechanical Design Process*, Ullman, D.G., McGraw-Hill, NY, 1992.

[Xerox 88] "Xerox Product Development Process," notes from Xerox, Webster, NY, 1988.

[Yakemovic 89] "The Capture of Design Rationale on an Industrial Development Project: Preliminary Report," Yakemovic, K.B., and J. Conklin, MCC Technical Number STP-279-89, July 14, 1989.

[Yakemovic 90] "Report on a Development Project Use of an Issue-Based Information System," Yakemovic, K.B., and J. Conklin, MCC Technical Number STP-247-90, June 1990.