Simon Szykman
e-mail: szykman@nist.gov

Ram D. Sriram
e-mail: sriram@cme.nist.gov

National Institute of Standards and Technology,
Manufacturing Systems Integration Division,
100 Bureau Drive, Stop 8263,
Gaithersburg, MD 20899-8263

William C. Regli
Department of Mathematics
and Computer Science,
Drexel University,
3141 Chestnut Street,
Philadelphia, PA 19104
e-mail: regli@drexel.edu

The Role of Knowledge in Next-generation Product Development Systems

Information technology has played an increasingly important role in engineering product development. Its influence over the past decade has been accelerating and its impact in the coming decade will undoubtedly be immense. This paper surveys several research areas relating to knowledge representation, capture and retrieval, which will have a growing influence on product development. Each of these areas could, on its own, provide sufficient material for an entire survey paper. Unlike traditional survey papers, this paper does not attempt to provide a comprehensive review of a field of research from its inception to the present. Rather, this paper aims to touch on a representative selection of recent developments in these influential technical areas. The paper provides perspectives into the kinds of technologies that are emerging from rapidly expanding fields of research, and discusses challenges that must be overcome to enable transition of these technologies into industry practice to support the next generation of product development software tools.
[DOI: 10.1115/1.1344238]

1 Introduction and Context

Since its advent, computer-aided design and manufacturing (CAD/CAM) has had an immeasurable impact on product development in engineering industry, and consequently on society as well. As a result of their significance, CAD/CAM technologies were identified by the National Academy of Engineering in 1989 (the NAE's 25th anniversary) as an outstanding engineering achievement over the preceding 25 years, and more recently by the American Society of Mechanical Engineers as one of the greatest technologies of the 20th century. As these technologies mature further, there is little doubt that their impact on product development will continue.

In the past, product development was often done within a single company by co-located design teams. In more recent years, there has been a shift in product development paradigms. The complexity of modern products means that a single designer or design team can no longer manage the complete product development effort. Developing products without sufficient expertise in a broad set of disciplines can result in extended product development cycles, higher development costs, and quality problems. Product development is being more often done collaboratively, by geographically and temporally distributed design teams. There is a high level of outsourcing, not only of manufacturing but also of actual product development efforts. Product development across companies, and even within a single company, is often done within a heterogeneous software tool environment. The Internet and intranets are supplanting paper and telephones as a means of exchanging product development information.

Designers are no longer merely exchanging geometric data, but more general knowledge about design and the product development process, including specifications, design rules, constraints, and rationale. Furthermore, this exchange of knowledge often crosses corporate boundaries. As the complexity of products increases and product development becomes more distributed, new software tools will begin to cover a broader spectrum of product development activities than do the traditional mechanical CAD systems. As design becomes increasingly knowledge-intensive and collaborative, the need for computational frameworks to enable engineering product development, by effectively supporting the formal representation, capture, retrieval and reuse of product knowledge, becomes more critical.

The vision held by some for future product development tools is that of a monolithic software system. In this vision, the product development process will be supported by a single integrated application suite. Such a tool would attempt to address the needs of the new product development paradigm, allowing teams that are potentially distributed geographically or across corporate boundaries to access tools and data at different phases of product development in order to produce a product. This vision, though not an uncommon one, has a number of drawbacks associated with it. In general, since a monolithic system is intended to be as complete a solution as possible, less emphasis is put on interoperability with other systems. As a result, distributed product development efforts must often standardize on a single vendor's tool suite in order to effectively exchange information. Because the cost of monolithic systems tends to be high, there is a barrier to collaboration with potential partners who use different tools. Furthermore, the high cost can completely price out many of the small and medium sized businesses that form a large segment of the industry community. And lastly, a monolithic system does not allow users to pick and choose among different tools to customize a set that will best suit their needs.

In our view, the ideal next-generation systems for product development will be those with which individual companies or teams involved in given product development activities can collaborate using a heterogeneous set of software tools, and still meaningfully exchange information and pass knowledge between various phases in the process. Assuming the interoperability barrier can be overcome, this vision avoids several of the disadvantages associated with a monolithic system. Companies would not be required to standardize on the same software platform in order to collaborate on product development. Smaller companies using individual software tools due to limited resources would still be able to play a role in a larger company's supply chain. Larger companies would be able to assemble what they consider to be the best suite of tools from a selection of existing software products (possibly from competing vendors), and would be able to migrate more easily to new tools, although some effort would still be involved in doing so.

The engineering infrastructure of the future will be one that is distributed and collaborative, and one that allows designers, process planners, manufacturers, clients, and other related domain personnel to effectively communicate and exchange knowledge

Contributed by the Information Science Tech. Committee for publication in the JOURNAL OF COMPUTING AND INFORMATION SCIENCE IN ENGINEERING. Manuscript received Sept. 2000; revised manuscript received Nov. 2000. Associate Editor S. Urban.

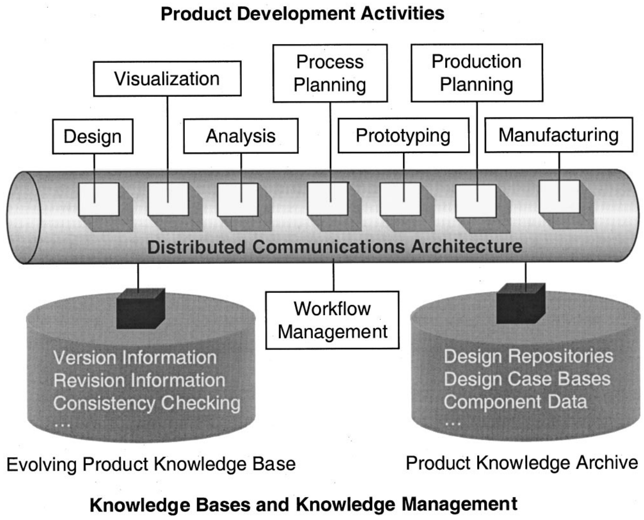

Fig. 1 High-level view of distributed product development

using a global information network. Figure 1 shows a high-level view of next-generation product development. The activities[1] shown at the top of the figure may be performed by one or more software applications, used by one or more development partners. The designers may be using heterogeneous systems, data structures, or information models, whose form and content may not be the same for all participants across all disciplines.

An extensive body of product development knowledge will be created and will evolve via interactions with various software applications. This knowledge will be stored in databases that will provide access to the evolving design knowledge at multiple levels of abstraction. Evolving product knowledge will be augmented by knowledge stored in design repositories, archives of corporate knowledge created to allow knowledge reuse in support of subsequent product development efforts.[2] Exchange of information among applications and partners, and access to evolving or archived knowledge will generally take place across some distributed communications architecture (e.g., web-based access or more general access via intranets or the Internet). Appropriate mechanisms are needed to realize the potential of this vision, be it for representing knowledge for effective exchange, formal capture of knowledge, or for allowing efficient retrieval of information to support knowledge reuse.

To successfully implement a computer-based distributed product development environment, requirements in four areas must be addressed: (1) software applications, to support product development activities, (2) standards, to support exchange of information among applications, (3) information technology (IT) infrastructure, on which the distributed environment will be built and will operate, and (4) organization, to address the changes in now product development is being done from the human perspective.

Technologies related to various product development activities and specific applications are among the topics of discussion in other papers appearing in this journal issue. A discussion of specific standards required to support interoperability and data exchange, both among CAD tools and between CAD tools and software tools used during other design activities can be found in [2]. Organizational issues, although less technical than the other areas listed above, are no less important. They are, however, outside the scope of this paper and are not discussed to any extent. The focus of this paper is primarily on the third of the areas mentioned—the information technology infrastructure. Specifically, this paper addresses the need for the integration of knowledge into this infrastructure, in order to enable a new generation of product development software tools that will better support the directions in which engineering industry is moving. This scope comprises, of course, only one portion of the overall IT infrastructure. Advances in the areas such as Internet-based communication architectures and general database technologies are also critical aspects of enabling the next generation of distributed product development environments.

The remainder of this paper focuses on the two sources of knowledge shown in Fig. 1: the evolving product knowledge base and the product knowledge archive. Section 2 addresses the issue of evolving product knowledge, providing discussions of knowledge representations and design rationale. Section 3 focuses on product knowledge archives, providing insights into technologies for capture, storage, and retrieval of knowledge to support knowledge reuse in product development. The paper provides perspectives into the kinds of technologies that are emerging from a rapidly expanding field of research. Section 4 reviews several challenges discussed in the paper that must be overcome to enable transition of these technologies into industry practice to support the next generation of product development software tools.

2 Evolving Product Knowledge

2.1 Product Knowledge Representation. As product development in industry changes, so too do the kinds of interactions among partners. In the past, interactions among supply chains were often limited to exchange of geometric CAD files with companies that supplied manufactured parts. In today's environment, the supply chain serves as more than a means to outsource manufacturing. Companies are outsourcing a greater portion of the product development process, often providing not detailed geometry to suppliers but requirements, specifications, interface definitions and constraints, leaving actual design activities to the lower tier suppliers.

While the current generation of product development software tools has provided extensive benefits among traditional product development processes, they do not adequately support the needs of industry's new paradigm described in the introduction. Currently, engineers *are* succeeding in exchanging information across distributed design teams and corporate boundaries earlier, and reusing information to a greater extent. But because existing software tools do not capture a broad spectrum of product development information, these exchanges occur informally (face-to-face across a table, by phone, by paper). It is a lack of formal representations for product development information that creates a significant barrier to its effective capture and exchange. Such representations have begun to emerge in various research efforts, but have yet to transition into commercial software systems. Industry has expressed a desire for standardization in anticipation of new capabilities within CAD systems, rather than in reaction to these capabilities after their commercial implementation. Industry needs include a more comprehensive representation of product knowledge than provided by existing representations, including concepts such as function, behavior, and others [3].

The use of function has long-since been recognized as an important part of the design process. Formalization of approaches to representing and reasoning about function, and using this knowledge to drive design, are in comparison relatively new in the engineering field. Much of the early research in the area function representation was performed by research in the field of artificial intelligence (AI). Even definitions of function have varied, indicating that the concept is a complex one. Function is often characterized in the literature as a relation between the input and output of energy, material, and information. Pahl and Beitz retain this

[1]The activities shown are intended to be representative, not comprehensive. A more complete list of activities has been developed as part of the SIMA (Systems Integration for Manufacturing Applications) Reference Architecture developed at NIST [1].

[2]In the figure, the presence of single databases representing evolving knowledge and for archived knowledge is for illustrative purposes and should not be taken literally. In reality, knowledge may be distributed among multiple sources, not all of which may be accessible to all software applications or to all development partners.

characterization but generalize the concept, defining function as an abstract formulation of a task, independent of any particular solution [4]. In the AI field, definitions of function have often involved the concept of behavior (e.g., [5–11] and others).

The variety in definitions of function has led to a variety of uses and representations of function. Baxter et al. distinguish two types of representations: models based on inputs and outputs of flows, and syntactic languages [12]. The first type generally follows the Pahl and Beitz paradigm of the flow of materials, energy, and signals through a hierarchy of functions. In some cases, function is mapped closely to physical elements in a system [13,14]. Other approaches view the functional description of a system as being described by an abstract functional decomposition that may, but need not, have a direct mapping onto an isomorphic physical decomposition of assemblies and subassemblies [8,12,15,16].

Syntactic languages describe a design artifact using a grammatical approach where a grammar is used to capture information about function. In general, these grammars consist of combinations of verbs (functions) and nouns (parts of a design artifact, or flows) such as ''transmit linear motion'' or ''create lateral motion'' [15,17–20]. Approaches such as these can capture the essence of the function of a given artifact. However, they do not fully address the needs of a formal representation of function because they lack formal information models that capture one or more other types of information relevant to function. Examples of such information include the explicit mappings between the function domain and the physical domain, in the form of links between functions and flows, as well as links between flows and their sources and destinations.

The engineering design community has been developing new classes of tools to support knowledge-based design, product data management (PDM), and concurrent engineering. When contrasted with traditional CAD tools, these new systems are making progress toward the next generation of engineering design support tools. However, these systems often focus primarily on database-related issues and do not place a primary emphasis on information models for artifact representation (e.g., [21–24]). Furthermore, although these systems can represent some kinds of non-geometric knowledge such as information about manufacturing process or bills of materials, representation of the artifact itself is still generally limited to geometry. The lack of a formal product representation that includes function, behavior, and structure has been identified as a shortcoming of existing PDM systems [25]. Research toward the development of more general artifact modeling representations has resulted in a significant body of work. Several of these efforts have focused on representation formats that provide a high level division into form, function, behavior. The division of design artifact knowledge into these categories has its roots in earlier work in intelligent design system development such as [26–29]. Work done in the design and engineering community includes [7,8,12,30–33] and others.

More recent efforts have focused specifically on the goal of developing of a generic representation of product knowledge that includes other kinds of product knowledge beyond form, function, and behavior, in order to support a broader level of information exchange and software tool interoperability [34]. Figure 2 shows some of the kinds of knowledge that might be part of such a representational infrastructure. In the figure, boxes within boxes denote compositional relationships: an artifact is composed of subartifacts and has functions, form, behaviors; form is composed of a combination of geometry and material knowledge, etc.[3]

One of the benefits of this kind of representation is that it supports design at earlier stages of the product development process by allowing designers to maintain a representation of a product at multiple levels of abstraction simultaneously. For instance, such a representation allows a function structure to be defined before any artifacts are created, or allows an artifact to exist in the absence of any specified geometry. Such a representation provides support for multiple levels of abstraction by allowing not only models of physical entities, but representation of concepts that are abstractions of physical entities. A representation like that shown in Fig. 2 also enables the decomposition of functions into subfunctions and the decomposition of artifacts into subartifacts. This supports multiple levels of abstractions in a different but equally important sense—that of allowing users to simultaneously represent a product at multiple levels of detail. So, for example, at one level a pump can be a single entity with its functions, but at another level it is represented as a collection of subartifacts (such as a motor), each having its own functions, and eventually each having its own form. Each subartifact can, in turn, be modeled as a single entity at one level, and as a collection of components at another. Supporting conceptual abstractions as well as multiple levels of abstraction will allow the development of tools that can be used starting at early stages of design rather than detailed design where most of today's tools are used.

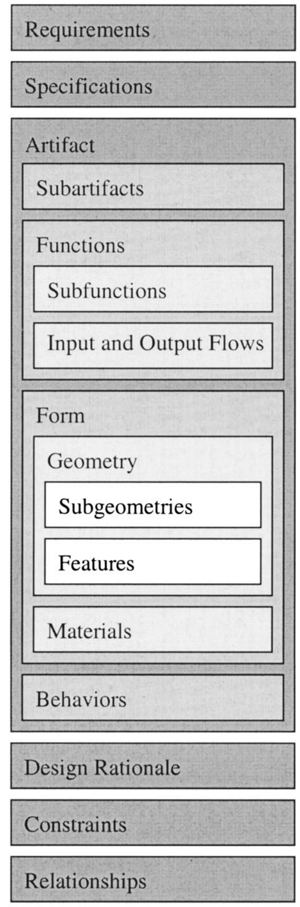

Fig. 2 Product knowledge representation

2.2 Design Rationale. Along with capturing a more comprehensive representation of artifacts in an evolving knowledge base, there is also a need to capture design rationale. Design rationale encompasses the broader context surrounding the product development process, including information about decisions, why they have been made, as well as relationships or dependencies that may either link decisions to part of the product representation (a function, artifact, etc.), or to other decisions. Because industry now frequently develops products using teams of individuals that are spread across corporate boundaries, capturing design rationale is an important issue in supporting an evolving product knowledge base.

[3] Other types of knowledge, such as configuration/version information, and the links that capture connections between the various knowledge entities that appear in the figure, are not explicitly shown but are expected to be part of a comprehensive product representation.

Documenting rationale or intent allows knowledge to flow even when there are corporate, geographic or temporal boundaries that inhibit person-to-person communication of information. More importantly, rationale capture can make explicit the fact that decisions were made, and why they were made, when that knowledge might otherwise be lost or available only to some of the people who are contributing to a product development effort. A decision can be influenced by prior decisions, and if properly captured, can also impact future decisions concerning the same product or subsequent ones. Thus, design rationale capture can reduce the time and/or cost associated with downstream efforts needed to resolve conflicts, or solve problems resulting from an inadequate understanding of interactions among design decisions.[4] Capturing design rationale has been a research topic for several decades. More comprehensive reviews of design rationale work can be found in [35,36].

Representing design rationale requires that one explicitly document the reasoning and argumentation in design [37]. There are two fundamental and complementary, representations of design rationale. First is the notion of design rationale as the recording of the design intent of an artifact. For example, in traditional mechanical design, rationale might include a functional description, geometric or assembly constraints and performance criteria (such as captured with one or more of the Product Knowledge Representation schema noted in Section 2.1). The second view is of design rationale as a record of the design process, the communications among agents, the decision-making that occurs, as well as the decision-making process. This view of rationale has been often studied in the software design and computer-supported collaborative work communities, where one goal is to support organizational intelligence and group decision making.

Recent work toward understanding the design process has provided a set of prototypes that show how to integrate design rationale with other design support tools such as CAD/CAM and CAE [38]. The majority of design rationale approaches fall into two categories: *process-oriented* and *feature-oriented*. For design in fields with a high degree of standardization, a *feature-oriented* approach to design rationale can describe a set of constraints and procedures that create the design; in more dynamic domains, a *process-oriented* design rationale provides a historical representation of the artifact [39,40].

Most design rationale systems aim to let designers explore design alternatives within a certain knowledge representation framework. Based on a representational schema for design rationale in the domain of interest, the decisions and reasoning are either recorded by designers themselves (manually entered data, stored documents, etc.) or are captured by an agent that monitors the design process and attempts to extract information automatically from sources such as electronic mail, archived design meetings, or designers' notebooks. Representation impacts the methods that can be used for capturing and retrieving design rational, so the choice of an appropriate representation is an important one.

In the literature, representations fall primarily into two categories: *argumentation-based* representations or *descriptive* representations. Argumentation-based techniques employ a semi-formal graphical format for laying out the structure of arguments where labeled nodes and edges connect design issues and relationships [40,41]. In argumentation-based systems, designers can maintain consistency in decision-making, keep track of decisions, and communicate with one another about design reasoning. Descriptive approaches record the history of design activities, workflow, and communication between designers. More specifically, they record what decision are made by designers, as well as when, by whom,

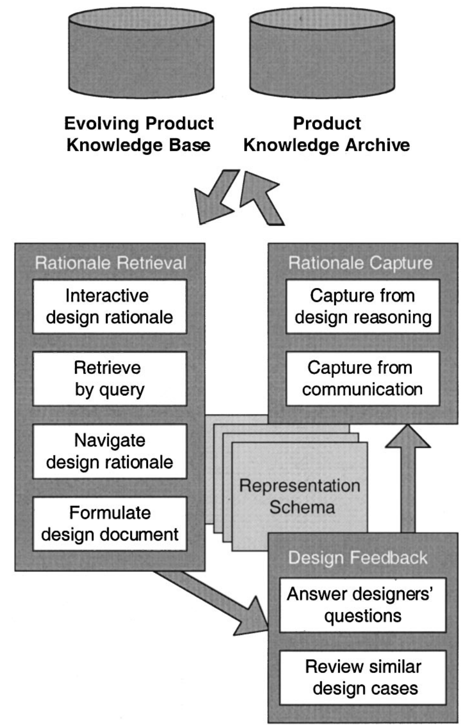

Fig. 3 Generic design rationale system architecture

and why they are made. This approach is used in dynamic design domains where there may be little or no standardization of solutions. Descriptive approaches emphasize minimizing the intrusion to the designers by making design rationale capture as transparent as possible.

Usually, design rationale capture consists of two steps: *knowledge recording* and *design rationale construction*. A generic architecture for a design rationale system is shown in Fig. 3. The first step is to record as much information as possible during the design process. The second step is to organize the rationale knowledge based on the representation schema and constraints in the design domain, and store this information in a knowledge base. The recording and construction processes have been implemented in design rationale systems as both by automatic capture and via user intervention (i.e., requiring designers themselves to input or record the design discussions, decisions, and reasoning). However, user intervention-based design rationale capture has met only with limited success, as designers are typically very reluctant to spend time on annotating their designs with rationale.

Some preliminary work has been done on developing automated design rationale capture tools. Garcia and Howard [42] focus on parametric design, incorporating a model of the design process in the HVAC (Heating Ventilation and Air Conditioning) domain. Ganeshan et al. [43] generate design rationale by using a transformational approach. Quereshi et al. [44] emphasize the development of a detailed product and process model in the electromechanical design area. Myers et al. [45] focus mainly on the detailed design stage, in particular re-design. Their system is built over a commercial CAD tool (Bentley's Microstation95[5]).

The interactions between various participants—often having diverse and/or opposing views—in a large-scale engineering project produces different types of conflicts. The use of design rationale to mitigate such conflicts is addressed by Klein's Design Ratio-

[4]The preceding discussion provides a strong motivation for including design rationale in an evolving product knowledge base. Although the topic of design rationale is not revisited in Section 3 of the paper, it should be noted that design rationale is as important for product knowledge archives as it is for evolving knowledge. If design rationale is captured as part of an evolving knowledge base, that rationale should transition into a product knowledge archive along with the rest of the knowledge about the product.

[5]Commercial software systems, many of which are registered or trademarked, are identified in order to provide implementational information. In no case is such identification a recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the tools identified are necessarily the best available for the given purpose.

nale Capture System [46] and Peña-Mora's SHARED-DRIMS [47]. Nexprise's commercial ipTeam™ system [48] provides a module called ConsensusBuilder™, which aids in design rationale capture during a collaborative design session.

3 Product Knowledge Archives

3.1 Knowledge Storage. Recently, commercial database system developers as well as academic and industry researchers have been investing in research for management of large multimedia databases [49,50]. This includes National Science Foundation Digital Libraries initiatives (DL-I and DL-II) [51] as well as commercial systems [52–54]. The commercial database vendors all have advanced architectures that include domain-specific layers built on top of a standard relational or object-oriented database. Some recent efforts from both academia and industry have studied how to intelligently handle engineering design data [53,55].

Most of the approaches toward database storage of CAD models parallel techniques for other multimedia applications. For example, a common approach is to take the CAD model (a solid model or surface model) represented in some neutral format (such as IGES [56] or ISO 10303, Standard for the Exchange of Product model data, commonly referred to as STEP [57]) and create a pictorial image or volume/voxel representation with which model indexing and retrieval is performed. These approaches do not consider information pertaining to manufacturing or design features, tolerances, or design knowledge that might be present in the corporate database, nor do they seem to be suitable for CAD assemblies, where inter-part relationships and models of function and behavior are much more important than gross shape properties. A related survey on geometric databases can be found in [58]. Specific to knowledge-based storage and retrieval of CAD data, Hardwick et al. [59] have merged databases with the Internet and STEP-based engineering standards. Over the last five years Regli and his colleagues have initiated the National Design Repository [60,61], the largest publicly accessible collection of online engineering designs and related engineering data.

The nature of large-scale engineering knowledge bases requires research and evaluation of complexity management techniques for efficient management of the complex and extensive sets of data elements associated with them. Computational operations on geometric information such as solid models are floating-point and memory intensive. Furthermore, many existing knowledge-based systems require time consuming intervention by people to input the necessary information. For large manufacturing companies in today's industry, in-house centralized knowledge bases of design and manufacturing data would require terabytes of memory storage and consist of millions of individual components and relationships. As a result, the need for algorithms that operate on this information to perform traditional database tasks such as indexing, matching, and retrieval, presents significant challenges to the research community.

Another emerging area of research is that of digital libraries. Part libraries and catalogs have been an area of active study and development by the standards community [57,62] and internal to organizations using product data management tools and databases. The emphasis in these efforts is often on the development of schemata for representing catalog information rather than on product development knowledge. Although a significant portion of components used in products are obtained from standard part catalogs rather than designed in-house, the kinds of knowledge that facilitate the process of integrating these parts into the context of the greater whole generally go beyond what is available in traditional catalog data. Merely providing access to schematics and CAD models is often inadequate for this purpose.

Some of industry's needs that are not satisfied by electronic part catalogs or traditional geometry-oriented corporate design databases are being addressed by the emerging research area of design repositories—repositories of heterogeneous knowledge and data that are designed to support representation, capture, sharing, and reuse of corporate design knowledge. It should be noted that although the term *design repositories* has not yet found its way into daily usage in industry, many companies are migrating from traditional design databases to what would be considered design repositories as characterized in this paper. Although design repositories can in general terms be thought of as design databases—and indeed will most often be implemented using database management systems—design repositories are distinguished from traditional design databases in several significant ways:

- Design repositories attempt to capture a more comprehensive product representation that traditional CAD databases, including the kinds of knowledge discussed previously (i.e., function, behavior, rationale, etc.). It should be noted, however, that a representation that fully encompasses every aspect of a design might simply not be possible.
- Design databases typically contain images (drawings), CAD models, and unstructured text (documentation). Design repositories tend to be more heterogeneous and may contain formal schemata and data structures, structured text, mathematical simulation models, animations, video, and other types of information.
- Design databases tend to be static sources of information (though their contents may grow with time). While they are used for storage and retrieval of design data, capabilities for supporting the design process are not traditionally built into these systems. Such capabilities might include search for components/assemblies that satisfy required functions, explicit representation of physical and functional decompositions and the mappings between them, (partially) automated reasoning about a design, and more. Since design databases have not been designed specifically for these purposes, they are limited in their ability to meet needs for design of large-scale engineering systems.

The NIST Design Repository Project is an ongoing project at the National Institute of Standards and Technology that involves research toward providing a technical foundation for the creation of design repositories [33]. The infrastructure being developed consists of formal representations for design artifact knowledge, and web-based interfaces for creating design repositories. Design repositories of the future will also provide not only an expanded representation of product knowledge and data, but also mathematical models that can be used to support virtual prototyping and evaluation via composable simulation of product behavior and performance. Given a simulation model and a method for defining generic interfaces for creating and integrating multiple models of this kind, complex systems could be simulated virtually rather than physically prototyped, allowing evaluation of many more design alternatives at early stages of product development.

Work in the latter area has been done as part of several projects including the Model-Based Support of Distributed Collaborative Design (previously How Things Work) project at the Stanford University Knowledge Systems Laboratory [10,63], the Active Catalog project at the University of Southern California Information Sciences Institute [64], and the Composable Simulation Project at Carnegie Mellon University [65]. Significant challenges remain in the successful integration of recent and future advances in knowledge representation, database technology, and composable simulation. Novel interfaces for browsing and authoring these complex product knowledge bases are also needed to attain the objectives embodied in the design repository concept.

3.2 Knowledge Retrieval. If we assume the existence of repositories of archived product knowledge, there are two facets to the problem of knowledge retrieval. The first is facilitating access to stored knowledge through the use of formal knowledge representations (discussed in Section 2.1), indexing techniques, and the issue of the role of standardized language and terminology. The second aspect of the problem is that of the actual mechanisms for retrieving the desired knowledge from a repository.

The need for standardized terminology in product development

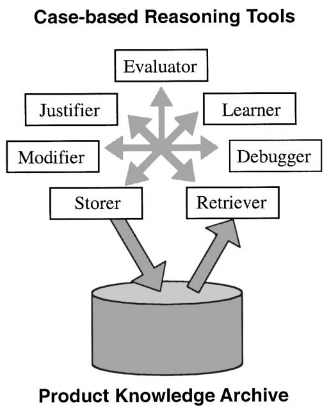

Fig. 4 Case-based reasoning system

software tools is often overlooked in the literature; however, it is an issue of critical importance for a number of reasons. The first reason is to reduce ambiguity at the modeling level. Ambiguities can occur when multiple terms are used to mean the same things, or when the same term is used with multiple meanings. The distillation of a large body of terms into concise taxonomies does not eliminate this problem entirely, but it significantly lessens its occurrence. A related issue is that of uniqueness, not at the level of individual terms as with synonyms, but at the concept level. The larger the number of terms there are in a vocabulary, the more ways there are to model or describe a given concept. This makes processing of information that has been represented more difficult, whether it be a human trying to interpret information modeled by somebody else, or algorithms developed for function-based reasoning or design automation. In practice, it is impossible to have a vocabulary that allows all concepts to be modeled, in only one unique way, because it is the flexibility required for representation of a broad set of concepts that results in multiple ways of expressing the same concept. However, to whatever extent uniqueness problems at the concept level can be reduced, interpreting information that is represented can be made easier. A third reason for developing a standardized terminology is that it increases the uniformity of information within function models. This will facilitate the exchange of function information among distributed researchers and developers, and will greatly simplify the task of indexing and retrieval of information for the purposes of function-based searches and query capabilities. Despite a large body of work in the area of ontologies, only a handful of efforts have focused specifically on domains related to design and product development. A more comprehensive discussion relating to ontologies for engineering is presented in [66], and is therefore not provided in this paper.

An area of research that is more directly relevant to the issue of retrieval of product knowledge from comprehensive knowledge repositories is that of case-based design and case-based reasoning. In the context of this body of work, the repositories of corporate product knowledge would serve as case bases. Figure 4 illustrates a set of components that comprise a case-based reasoning system. Common components of such systems include a retriever for retrieving previous cases, a storer for storing new or modified cases, an evaluator for ranking retrieved cases, and a modifier for adapting a retrieved case. Other possible components include a justifier for simulating or rationalizing a modified solution, a learner for learning general solution strategies, and a debugger for determining causes of failed attempts at solutions.

In the research community, case-based reasoning research generally focuses on automating the synthesis of new solutions based on previous ones. In practice, there may not always be an intent, or even an ability, to use archived product knowledge to generate new design solutions automatically. Nevertheless, whether the goal is automated synthesis or computer-supported designer synthesis, the basic need to retrieve and reuse knowledge in subsequent design activities remains the same. From this perspective, the common requirements are to take as input some articulation of a target design or specification, to retrieve previously-generated knowledge according to some measure of similarity, and to evaluate multiple potential matches to determine which most closely meets the specification.

Prior work in the area of case-based reasoning has resulted in a foundation of structures, algorithms, and techniques for reasoning about and adapting archived knowledge (e.g., [67–72] and others). Engineering design and manufacturing have been areas of considerable interest among researchers in this field. CADET [73] and its descendent projects focused on conceptual design, solving problems by the use of relationships that capture function, structure, and behavior contained in the system's knowledge base. Case retrieval is performed using variations of graph matching, and is supported at varying degrees of abstraction. A framework for a case-based reasoning system using CADET as a case study is provided in [74]. KRITIK and its descendent INTERACTIVE KRITIK [28] operate on design problems using a case-base of designs represented by symbolic component descriptions, their relationships and behaviors. A central contribution of KRITIK was the formalization of a structure-behavior-function model for designs, where design cases can be indexed according to the functions they deliver. Although KRITIK's successor systems have extended many of the earlier concepts, the reasoning techniques are still primarily symbolic and have not been coupled with detailed engineering data. Other case-based reasoning tools in engineering domains include systems for assembly and assembly planning [75,76], architecture [77–79], civil engineering [80,81], the application of the TOLTEC planner to manufacturing problems [82], the design of planar linkage assemblies [83], and numerous others.

Recent issues of IEEE Expert and Intelligent Systems have emphasized past accomplishments and current challenges in extending AI and case-based reasoning to complex engineering problems [84,85]. This series provides several observations, including (1) that while there has been much research in knowledge-based engineering systems, the integration of this research into existing CAD tools has yet to really begin, and (2) that existing research approaches still have great difficulty scaling to complex design cases. In a survey of recent work on variant and case-based design, Fowler [86] notes that better abstract models are needed for mechanical artifacts so that function information can be stored in the CAD knowledge base. The implication of these observations is that there is a disconnect between recent progress in engineering product representation and in case-based reasoning systems.

4 Looking Forward

This paper surveys several research areas relating to the use of knowledge and its increasing impact on product development. The paper provides perspectives into the kinds of technologies that are emerging from rapidly expanding fields of research, and discusses challenges that must be overcome to enable transition of these technologies into industry practice, to support the next generation of product development software tools. The advent of the Internet is in the process of making collaborative design paradigms pervasive. As companies under pressure to reduce costs and development times work to realize the design-anywhere manufacture-anywhere concept, a company's ability to compete globally will depend upon how well it can leverage existing knowledge sources. Mechanisms need to be developed for encoding, indexing, retrieving, and using this knowledge. A research agenda for development of technologies to bring about this vision should address the following challenges:

Development of a comprehensive representation for product development knowledge. Several of the representational frameworks described in Section 2.1 include information that goes beyond what is captured in traditional CAD tools, such as representation of function, form, physical decomposition of an artifact into subassemblies and components, and mappings between the physical domain and the function domain. Nevertheless, they all have gaps in their representational ability. For instance, none yet provide a detailed assembly model; information about the kinds of part matings, kinematic joints, and assembly constraints is not currently captured, nor is tolerance information (at either the part level or the assembly level). Although design rationale has been the subject of numerous research efforts, the explicit representation of design rationale as part of a comprehensive product knowledge representation is an issue that must still be addressed. It is also desirable to integrate a comprehensive product model with a more formal representation of behavior that will enable composable simulation for evaluation of a product as it evolves, as discussed briefly in Section 3.1.

Integration of traditional engineering software with knowledge-based applications. Among the numerous manufacturing research challenges recently identified by the National Research Council [87], embedding knowledge into manufacturing systems is universal to all the research agenda items identified. Efforts, such as the ISO 10303 (STEP) standard have primarily focused on standardizing the exchange of data. In this area, progress is being made toward extending standards to address gaps in their coverage. Various organizations and standards committees are developing representations for data associated with assembly models, tolerances, and parametric information, among others.

Beyond the issue of engineering *data*, there is for need for representations that capture various types of *knowledge* used in knowledge-intensive engineering activities. Such representations can lay a foundation for the creation of engineering ontologies, thus aiding in the development of testable knowledge exchange mechanisms where agents—both human and computer —can exchange specific design and manufacturing knowledge in a meaningful way. Descriptions of progress in this area are discussed in [67]. Further work along these lines will facilitate the exchange of knowledge both among knowledge-based systems as well as between knowledge-based systems and traditional CAD systems. This research will also facilitate the integration of software applications that have not been explicitly designed to interoperate with one another.

Mechanisms for indexing, searching, and retrieving design cases. To integrate retrieval and evaluation (and more generally case-based reasoning) techniques with next-generation product development systems requires advances that specifically target the nature of product development data and knowledge. CAD data is geometry-intensive and includes engineering data that are difficult to give formal structure to, such as tolerances or design and manufacturing feature information. Additional product knowledge may include textual descriptions, design rules, functional models, and other kinds of information.

Research efforts in the area of case representation are needed to develop semantic structures that can be used to unite the geometric representations of solid modeling, formal representations of other kinds of product knowledge (function, behavior, etc.), and the dominantly symbolic representations used in case-based reasoning and design systems. In addition to developing methods that are well suited to these different kinds of knowledge, indexing schemes must be able to support some level of automation as well. While knowledge resulting from the use of next-generation software tools may be more comprehensive and more formally represented than the current generation of tools, there is a large body of legacy data that currently exists which cannot be abandoned or ignored.

To date, much of the research in the area of retrieval mechanisms that go beyond traditional text-based techniques has centered on retrieval of 2-D images and 3-D geometry. This has been possible because the representations used for this subset of engineering data are relatively mature. Although there have been efforts that address retrieval for other types of engineering knowledge (several of which have been cited in this paper), these techniques will only find their way into widespread use if the knowledge to which they are applied is consistently available in some pre-known format. This issue reinforces the need for consensus among product development software developers and users regarding the content and form for more comprehensive representations of engineering knowledge.

Design rationale capture and conflict mitigation. The multiple, often competing, perspectives of participants in a collaborative product development effort typically lead to many conflicts. These conflicts, if not resolved early, create more costly designs, delays in the development process, and compromises in the final product. Thus, a fundamental issue in collaborative engineering is conflict mitigation. Conflicts often stem from a lack of information that certain specialists have about other specialists' objectives, and reasons for rejecting or accepting a given alternative (i.e., design rationale). On the other hand, if the design rationale of all participants is made available to everyone, designers can become overwhelmed with complex information. Thus, there is a pressing need for systems that help designers capture, interpret, and easily utilize this data when required for conflict resolution.

Requirements for such a system include the ability to capture and represent the evolution of design intent, knowledge about the artifact as it evolves through the product development process, and relationships that may either link decisions to the product (an artifact, a function, etc.) or to other decisions. A primary issue that must be addressed is how design rationale is to be acquired. The capture of design rationale should be as unobtrusive as possible. On the other hand, it is not clear that rationale capture can be fully automated so it is likely that some of the burden will fall on the designers who must supply their rationale in order for it to be recorded. A more detailed discussion of the issues associated with conflict mitigation, negotiation support tools, and approaches to automating negotiation, are beyond the scope of this paper; however, work is being undertaken in the research community to address these needs.

Technology transfer. Ultimately, in order for any of this work to have an impact on the engineering industry, it must transition out of the research community and into use by industry. There are three ways in which this technology diffusion can occur. First, industry engineers can read papers and/or books that describe the use of these technologies, and use that knowledge to improve their own design processes. Second, engineers in industry can develop product development software tools that incorporate advanced technologies such as those described in this paper. Finally, commercial CAD/CAM/CAE (computer-aided design/manufacturing/engineering) software tool vendors can begin to incorporate these technologies into their next-generation commercial offerings.

Ultimately, the greatest impact will result from diffusion into commercial software systems, but this diffusion will not happen without pressure from the user community in industry. A precursor to successful diffusion of technology is the gathering of feedback and buy-in from both researchers and industry end-users. Computer-aided engineering technologies have had an enormous impact in engineering industry, and consequently on society as well. Information technology continues to play an increasing role in product development every day. As these technologies mature in the coming years, they will enable capabilities that were not conceivable even a decade ago.

References

[1] Barkmeyer, E., Christopher, N., Feng, S., Fowler, J., Frechette, S., Jones, A., Jurrens, K., McLean, C., Pratt, M., Scott, H. A., Senehi, M. K., Sriram, R. D., and Wallace, E., 1997, *SIMA Reference Architecture Part I: Activity Models*, NISTIR 5939, National Institute of Standards and Technology, Gaithersburg, MD.

[2] Sriram, R. D., 1999, *Standards for the Collaborative Design Enterprise*, Response to Object Management Group (OMG) Manufacturing Domain Task Force Request for Information MfgDTF RFI#4, available online at ⟨URL:http://cgi.omg.org/cgi-bin/doc?mfg/99-08-04.pdf⟩.

[3] Szykman, S., Sriram, R. D., and Smith, S. J., (eds.), 1998, ''Breakout Session #1—Industry Perspective,'' ''Breakout Session #2— Database/Knowledge-Base Frameworks,'' ''Breakout Session #3—Standards for Information Modeling/Knowledge Representation,'' *Proceedings of the NIST Design Repository Workshop*, Gaithersburg, MD, November 1996, pp. 39–49.
[4] Pahl, G., and Beitz, W., 1988, *Engineering Design: A Systematic Approach*, Springer-Verlag, New York.
[5] de Kleer, J., 1984, ''How Circuits Work,'' Artif. Intell., **24**, pp. 205–280.
[6] Chandrasekaran, B., 1994, ''Functional Representation: A Brief Historical Perspective,'' Appl. Artif. Intell., **8**, pp. 163–197.
[7] Qian, L., and Gero, J. S., 1996, ''Function-Behavior-Structure Paths and Their Role in Analogy-Based Design,'' Artif. Intell. Eng. Des. Anal. Manufact., **10**, No. 4, pp. 289–312.
[8] Umeda, Y., Ishii, M., Yoshioka, M., Shimomura, Y., and Tomiyama, T., 1996, ''Supporting Conceptual Design Based on the Function-Behavior-State Modeler,'' Artif. Intell. Eng. Des. Anal. Manufact., **10**, pp. 275–288.
[9] Prabhakar, S., and Goel, A. K., 1998, ''Functional Modeling for Enabling Adaptive Design of Devices for New Environments,'' *Artificial Intelligence in Engineering*, **12**, pp. 417–444.
[10] Iwasaki, Y., Farquhar, A., Fikes, R., and Rice, J., 1997, ''A Web-Based Compositional Modeling System for Sharing of Physical Knowledge,'' *Proceedings of the 15th International Joint Conference on Artificial Intelligence*, AAAI Press, pp. 494–500.
[11] Sasajima, M., Kitamura, Y., Ikeda, M., and Mizoguchi, M., 1996, ''Representation Language for Behavior and Function: FBRL,'' Exp. Syst. Appl., **10**, No. 3/4, pp. 471–479.
[12] Baxter, J. E., Juster, N. P., and de Pennington, A., 1994, ''A Functional Data Model for Assemblies Used to Verify Product Design Specifications,'' Proc. IMechE, Part B, **208**, No. B4, pp. 235–244.
[13] Liu, Y. C., Chakrabarti, A., and Bligh, T. P., 1999, ''Transforming Functional Solutions into Physical Solutions,'' *Proceedings of the 1999 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Paper No. DETC99/DTM-8768, Las Vegas, NV, September.
[14] Shapiro, V., and Voelcker, H., 1989, ''On the Role of Geometry in Mechanical Design,'' Res. Eng. Des., **1**, pp. 69–73.
[15] Kirschman, C. F., and Fadel, G. M., 1998, ''Classifying Functions for Mechanical Design,'' ASME J. Mech. Des., **120**, No. 3, pp. 475–482.
[16] Szykman, S., Racz, J. W., and Sriram, R. D., 1999, ''The Representation of Function in Computer-based Design,'' *Proceedings of the 1999 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Paper No. DETC99/DTM-8742, Las Vegas, NV, September.
[17] Lai, K., and Wilson, W. R. D., 1989, ''FDL—A Language for Function Description and Rationalization in Mechanical Design,'' ASME J. Mech., Transm., Autom. Des., **111**, pp. 117–123.
[18] Hundal, M. S., 1990, ''A Systematic Method for Developing Function Structures, Solutions and Concept Variants,'' Mech. Mach. Theory, **25**, No. 3, pp. 243–256.
[19] Sturges, R. H., O'Shaughnessy, K., and Kilani, M. I., 1996, ''Computational Model for Conceptual Design Based on Extended Function Logic,'' Artif. Intell. Eng. Des. Anal. Manufact., **10**, pp. 255–274.
[20] Stone, R. B., and Wood, K. L., 1999, ''Development of a Functional Basis for Design,'' *Proceedings of the 1999 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Paper No. DETC99/DTM-8765, Las Vegas, NV, September.
[21] Bliznakov, P. I., Shah, J. J., and Urban, S. D., 1996, ''Integration Infrastructure to Support Concurrence and Collaboration in Engineering Design,'' *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper No. 96-DETC/EIM-1420, Irvine, CA, August.
[22] Hardwick, M., and Loffredo, D. 1995, ''Using EXPRESS to Implement Concurrent Engineering Databases,'' *Proceedings of the 1995 ASME Computers in Engineering Conference and Engineering Database Symposium*, Boston, MA, September, pp. 1069–1083.
[23] Shah, J. J., Jeon, D. K., Urban, S. D., Bliznakov, P., and Rogers, M., 1996, ''Databased Infrastructure for Supporting Engineering Design Histories,'' Comput.-Aided Des., **28**, No. 5, pp. 347–360.
[24] Wood, W. H., III, and Agogino, A. M., 1996, ''Case-Based Conceptual Design Information Server for Concurrent Engineering,'' Comput.-Aided Des., **28**, No. 5, pp. 361–370.
[25] Bilgic, T., and Rock, D., 1997, ''Product Data Management Systems: State-of-the-Art and the Future,'' *Proceedings of the 1997 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper No. DETC97/EIM-3720, Sacramento, CA, September.
[26] de Kleer, J., and Brown, J. S., 1983, ''Assumptions and Ambiguities in Mechanistic Mental Models,'' *Mental Models*, D. Gentner, and A. L. Stevens, eds., Lawrence Erlbaum Associates, Hillsdale, NJ, pp. 155–190.
[27] Chandrasekaran, B., Goel, A., and Iwasaki, Y., 1993, ''Functional Representation as Design Rationale,'' IEEE Comput., pp. 48–56.
[28] Goel, A., Gomez, A., Grue, N., Murdock, J. W., Recker, W., and Govindaraj, T., 1996, ''Explanatory Interface in Interactive Design Environments,'' *Artificial Intelligence in Design '96*, J. S. Gero, ed., Kluwer Academic Publishers, Boston, pp. 387–404.
[29] Alberts, L. K., and Dikker, F., 1992, ''Integrating Standards and Synthesis Knowledge Using the YMIR Ontology,'' *Artificial Intelligence in Design '94*, J. S. Gero and F. Sudweeks, eds., Kluwer Academic Publishers, Boston, pp. 517–534.
[30] Gorti, S. R., Gupta, A., Kim, G. J., Sriram, R. D., and Wong, A., 1998, ''An Object-Oriented Representation for Product and Design Process,'' Comput.-Aided Des., **30**, No. 7, pp. 489–501.
[31] Henson, B., Juster, N., and de Pennington, A., 1994, ''Towards an Integrated Representation of Function, Behavior and Form,'' *Computer Aided Conceptual Design, Proceedings of the 1994 Lancaster International Workshop on Engineering Design*, J. Sharpe and V. Oh, eds., Lancaster University EDC, Lancaster, pp. 95–111.
[32] Ranta, M., Mäntylä, M., Umeda, Y., and Tomiyama, T., 1996, ''Integration of Functional and Feature-Based Product Modeling—the IMS/GNOSIS Experience,'' Comput.-Aided Des., **28**, No. 5, pp. 371–381.
[33] Szykman, S., Racz, J. W., Bochenek, C., and Sriram, R. D., 2000, ''A Web-based System for Design Artifact Modeling,'' Des. Stud., **21**, No. 2, pp. 145–165.
[34] Szykman, S., Fenves, S. J., Shooter, S. B., and Keirouz, W., 2000, ''A Foundation for Interoperability in Next-generation Product Development Systems,'' *Proceedings of the 1999 ASME Design Engineering Technical Conferences and Computers and Information in Engineering Conference*, Paper No. DETC2000/CIE-14622, Baltimore, MD, September
[35] Moran, T., and Carroll, P., (eds.), 1996, *Design Rationale: Concepts, Techniques, and Use*, Lawrence Erlbaum Associates, Hillsdale, NJ.
[36] Lee, J., 1997, ''Design Rationale Systems: Understanding the Issues,'' IEEE Expert, **12**, No. 3, pp. 78–84.
[37] MacLean, A., Young, R., Belloti, V., and Moran, T., 1991, ''Questions, Options, and Criteria: Elements of Design Space Analysis,'' Human Comput. Int., **6**, pp. 201–250.
[38] King, J. M. P., and Banares-Alcantara, R., 1997, ''Extending the Scope and Use of Design Rationale,'' Artif. Intell. Eng. Des. Anal. Manufact., **11**, No. 2, pp. 155–167.
[39] Conklin, J. E., and Burgess-Yakemovic, K. C., 1991, ''A Process-Oriented Approach to Design Rationale,'' Human-Comput. Int., **6**, No. 3&4, pp. 357–391.
[40] Buckingham-Shum, S. J., and Hammond, N., 1994, ''Argumentation-based Design Rationale: What Use at What Cost?,'' Int. J. Human Comput. Stud., **40**, No. 4, pp. 603–652.
[41] Shipman, F. M., III, and McCall, R. J., 1997, ''Integrating Different Perspectives on Design Rationale: Supporting the Emergence of Design Rationale from Design Communication,'' Artif. Intell. Eng. Des. Anal. Manufact., **11**, No. 2, pp. 141–154.
[42] Garcia, A. C. B., and Howard, H. C., 1992, ''Acquiring Design Knowledge Through Design Decision Justification,'' Artif. Intell. Eng. Des. Anal. Manufact., **6**, No. 1, pp. 59–71.
[43] Ganeshan, R., Garrett, J., and Finger, S., 1994, ''A Framework for Representing Design Intent,'' Des. Stud., **15**, No. 1, pp. 59–84.
[44] Quereshi, S., Shah, J., Urban, S. D., Harter, E., Parazzoli, C., and Bluhm, T., 1997, ''Integration Model to Support Archival of Design History in Databases,'' *Proceedings of the 1997 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper No. DETC97/DTM-3876, Sacramento, CA, September.
[45] Myers, K. L., Zumel, N. B., and Garcia, P. E., 2000, ''Acquiring Design Rationale Automatically,'' Artif. Int. Eng. Des. Anal. and Manufact., **14**, No. 2, pp. 115–135.
[46] Klein, M., 1992, ''DRCS: An Integrated System for Capture of Designs and Their Rationale,'' *Artificial Intelligence in Design '92*, J. Gero, ed., Kluwer Academic Publishers, Boston, pp. 393–412.
[47] Peña-Mora, F., Sriram, R. D., and Logcher, R., 1995, ''Conflict Mitigation System for Collaborative Engineering,'' Artif. Intell. Eng. Des. Anal. Manufact., **9**, No. 2, pp. 101–124.
[48] Nexprise, 2000, Nexprise web site, ⟨URL:http://www.nexprise.com⟩.
[49] Wiederhold, G., 1995, ''Digital Libraries, Value, and Productivity,'' Commun. ACM, **38**, No. 4, pp. 85–96.
[50] Picard, R. W., and Pentland, A. P., 1995, ''Introduction to the Special Section on Digital Libraries: Representation and Retrieval,'' IEEE Trans. Pattern Anal. Mach. Intell., **18**, No. 8, p. 769.
[51] NSF, 2000, NSF Digital Libraries Initiative (Phase 2) Home Page, ⟨URL:http://www.dli2.nsf.gov/⟩
[52] Stix, G., 1997, ''Finding Pictures on the Web,'' Sci. Am., **276**, No. 3, pp. 54–55.
[53] Gupta, A., and Jain, R., 1997, ''Visual Information Retrieval,'' Commun. ACM, **40**, No. 5, pp. 71–79.
[54] Flickner, M., Sawhney, H. S., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., and Yanker, P., 1995, ''Query by Image and Video Content—The QBIC System,'' IEEE Trans. Comput., **28**, No. 9, pp. 23–32.
[55] Cybenko, G., Bhasin, A., and Cohen, K. D., 1997, ''Pattern Recognition of 3D CAD Objects: Towards an Electronic Yellow Pages of Mechanical Parts,'' Smart Eng. Syst. Des., **1**, pp. 1–13.
[56] ANSI, 1996, *Initial Graphics Exchange Specification IGES 5.3*, Document Number ANSI/US PRO/IPO 100-1996.
[57] ISO 13584-31:1999, 1999, *Industrial Automation Systems and Integration—Parts Library—Part 31: Implementation Resources: Geometric Programming Interface*.
[58] Kemper, A., and Wallrath, M., 1987, ''An Analysis of Geometric Modeling in Database Systems,'' ACM Comput. Surv., **19**, No. 1, pp. 1–45.
[59] Hardwick, M., Rando, T., Morris, K., Denno, P., and Spooner, D., 2000, ''Les-

sons Learned Developing Protocols for the Industrial Virtual Enterprise,'' Comput.-Aided Des., **32**, No. 2, pp. 159–166.
[60] Regli, W. C., and Cicirello, V. A., 2000, ''Digital Libraries for Engineering, Design and Manufacturing,'' Comput.-Aided Des., **32**, No. 2, pp. 119–132.
[61] Tuttle, R., Little, G., Regli, W. C., Corney, J., and Clark, D. E. R., 1997, ''Common Libraries for Networked Engineering Applications,'' *EUROPIA97: The Proceedings of the Sixth International Conference on Applications of Computer Networking in Architecture, Construction, Design, Civil Engineering, and Urban Planning*, Edinburgh, Scotland, April 2-3, pp. 1–11.
[62] Pierra, G., and Wiedmer, H. U., 1996, *Industrial Automation Systems and Integration—Parts Library—Part 42: Methodology for Structuring Part Families*, Technical Report ISO DIS 13584-42, International Organization for Standardization, May, ISO/TC 184/SC4/WG2 N243.
[63] Chouiery, B. Y., McIlraith, S., Iwasaki, Y., Loeser, T., Neller, T., Engelmore, B., and Fikes, R., 1998, ''Thoughts on a Practical Theory of Reformulation for Reasoning about Physical Systems,'' *Proceedings of the Symposium on Abstraction, Reformulation, and Approximation (SARA '98)*, pp. 25–36.
[64] Coutinhi, M., Eleish, R., Kim, J., Kumar, V., Ling, S. R., Neches, R., and Will, P., 1998, ''Active Catalogs: Integrated Support for Component Engineering,'' *Proceedings of the 1998 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper No. DETC98/CIE-5521, Sacramento, CA, September.
[65] Paredis, C. J. J., Diaz-Calderon, A., Sinha, R., and Khosla, P. K., 2001, ''Composable Models for Simulation-based Design,'' Engineering with Computers (to appear).
[66] Aamodt, A., and Plazas, E., 1994, ''Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches,'' AI Commun., **7**, No. 1, pp. 39–52.
[67] Ciocoiu, M., Gruninger, M., and Nau, D., 2000, ''Ontologies for Engineering Applications,'' ASME J. Comput. Inf. Sci. Eng., **1**, No. 1.
[68] Kolodner, J. L., 1993, *Case-Based Reasoning*, Morgan Kaufmann Publishers, San Mateo, California.
[69] Leake, D. B., (ed.), 1996, *Case-Based Reasoning: Experiences, Lessons, and Future Directions*, The AAAI Press/The MIT Press.
[70] Maher, M. L., and Pu, P., (eds.), 1997, *Issues and Applications of Case-based Reasoning in Design*, Lawrence Erlbaum Associates, Hillsdale, NJ.
[71] Slade, S., 1991, ''Case-Based Reasoning: A Research Paradigm,'' AI Mag., **12**, No. 1, pp. 42–55.
[72] Bardasz, T., and Zeid, I., 1991, ''Applying Analogical Problem Solving to Mechanical Design,'' Comput.-Aided Des., **23**, No. 3, pp. 202–212.
[73] Sycara, K., NavinChandra, D., Guttal, R., Koning, J., and Narasimhan, S., 1992, ''CADET: A Case-based Synthesis Tool for Engineering Design,'' Int. J. Exp. Syst., **4**, No. 2, pp. 157–188.
[74] Sriram, R. D., 1997, *Intelligent Systems for Engineering: A Knowledge-based Approach*, Springer-Verlag, New York.
[75] Kim, T. S., Han, S.-H., and Shin, Y. J., 1996, ''Product Data Management Using AP203 of STEP Standard,'' *Proceedings of the 1996 ASME Design Engineering Technical Conferences and Computers in Engineering Conference*, Paper No. 96-DETC/DAC-1069, Irvine, CA, August.
[76] Pu, P., and Reschberger, M., 1991, ''Assembly Sequence Planning Using Case-Based Reasoning Techniques,'' Knowledge-Based Syst., **4**, No. 3, pp. 123–130.
[77] Domeshek, E., and Kolodner, J., 1997, ''The Designer's Muse,'' *Issues and Applications of Case-based Reasoning in Design*, M. L. Maher and P. Pu, eds., Lawrence Erlbaum Associates, Hillsdale, NJ.
[78] Hua, K., and Faltings, B., 1993, ''Exploring Case-Based Building Design—Cadre,'' Artif. Intell. Eng. Des. Anal. Manufact., **7**, No. 2, pp. 135–143.
[79] Voss, A., 1997, ''Case Design Specialists in Fabel,'' in *Issues and Applications of Case-based Reasoning in Design*, M. L. Maher, and P. Pu, eds., pp. 11–38, Lawrence Erlbaum Associates, Hillsdale, NJ.
[80] Maher, M. L., Balachandran, M. B., and Zhang, D. M., 1995, *Case-Based Reasoning in Design*, Lawrence Erlbaum Associates, Hillsdale, NJ.
[81] Maher, M. L., and Gomez de Silva, A., 1996, ''Developing Case-based Reasoning for Structural Design,'' IEEE Exp. Intell. Sys., **11**, No. 3, pp. 42–52.
[82] Tsatsoulis, C., and Kashyap, R. L., 1993, ''Case-based Reasoning and Learning in Manufacturing with the TOLTEC Planner,'' IEEE Trans. Syst. Man Cybern., **23**, No. 4, pp. 1010–1023.
[83] Bose, A., Gini, M., and Riley, D., 1997, ''A Case-Based Approach to Planar Linkage Design,'' Artif. Intell. Eng., **11**, No. 2, pp. 107–119.
[84] IEEE, 1997, IEEE Exp. Intell. Syst., **12**, No. 2 (Special issue on AI in design).
[85] IEEE, 1997, IEEE Exp. Intell. Syst., **12**, No. 3 (Special issue on AI in design).
[86] Fowler, J. E., 1996, ''Variant Design for Mechanical Artifacts: A State-of-the Art Survey,'' Eng. Comput., **12**, pp. 1–15.
[87] NRC, 1995, *Information Technology for Manufacturing: A Research Agenda*, National Research Council, Washington, DC.