To appear in *Advances in Soft Computing – Engineering Design and Manufacturing*, Springer-Verlag, London, 1998

The NIST Design Repository Project

Simon Szykman, Ram D. Sriram,
Christophe Bochenek and Janusz Racz

National Institute of Standards and Technology
Manufacturing Systems Integration Division
Building 304, Room 6
Gaithersburg, MD 20899
szykman@cme.nist.gov, sriram@cme.nist.gov,
bchrist@cme.nist.gov and jwracz@cme.nist.gov

Abstract

Modern engineering industry is relying more and more on the use of knowledge in product development. This paper advocates design repositories as a natural progression from traditional design databases to systems that are created to more actively support knowledge-based design. In contrast to traditional design databases, design repositories serve not only as archives, but as repositories of heterogeneous information that are designed to enable representation, capture, sharing, and reuse of corporate design knowledge. This paper describes the NIST Design Repository Project, an ongoing project within the Engineering Design Technologies Group at the National Institute of Standards and Technology (NIST). The project objectives are to develop a computational framework for the creation of design repositories, and a proof-of-concept prototype to demonstrate their benefits. A number of research issues associated with the envisioned role of design repositories in industry are addressed. The current state of the project and its implementation are presented.

1 Introduction

Design of complex engineering systems is increasingly becoming a collaborative task among designers or design teams that are physically, geographically, and temporally distributed. 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. On the other hand, ensuring comprehensive technical proficiency in a world where trends are toward more multidisciplinary design can become a costly undertaking for a company.

Driven by such issues, companies are increasingly staffing only their core competencies in-house and depending on other firms to provide the complementary design knowledge and design effort needed for a complete product. Designers are no longer merely exchanging geometric data, but more general knowledge about design and design process, including specifications, design rules, constraints, rationale, etc. As design becomes increasingly knowledge-intensive and collaborative, the need for computational design frameworks to support the representation and use of knowledge among distributed designers becomes more critical.

In addition to sharing and exchanging information, pressure to reduce product development times has resulted in an increased focus on methods for representing and storing engineering artifact knowledge in a way that facilitates its retrieval and subsequent reuse. Merely providing access to schematics and computer-aided design (CAD) models of artifacts is inadequate for this purpose. In order to support reuse of engineering knowledge, a representation must convey additional information that answers not only "what?" questions about a design, but also "how?" and "why?" questions. Mappings from form to function have often been pointed to as just one example of the kind of information that is needed for effective reuse of design knowledge, but which is absent from traditional CAD models. The emerging research area of design repositories is aimed at addressing these industry needs. Design repositories make use of research in knowledge-based design to facilitate the representation, capture, sharing, and reuse (search and retrieval) 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 design repositories. While companies may still be referring to their corporate knowledge stores as design databases, in many cases these stores would fall under the definition of 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:

- Traditional design databases are typically more data-centric than knowledge-centric, and contain only a limited representation of an artifact such as drawings and/or CAD models, version information, and often related documentation. Design repositories attempt to capture a more complete design representation that may include characterization of function, behavior, design rules, simulation models, and so on. It should be noted, however, that a fully comprehensive representation of every aspect of a design may simply not be possible.
- Design databases are generally more homogeneous in the kinds of information they contain. In addition to containing images (drawings), file pointers (to CAD models), and unstructured text (documentation), design repositories may contain formal data/information models, structured text (specialized languages

for representing function, design rules, logical expressions), 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 built into traditional database 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, simulation of behavior and performance, (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.

This paper describes the NIST Design Repository Project, which has the objective of developing a framework to enable the creation of design repositories. This project is part of a larger effort called the Engineering Design and Process Planning Testbed, being conducted in the Manufacturing Systems Integration Division of the Manufacturing Engineering Laboratory at NIST. The following section describes the project objectives and identifies a set of related research issues. Within the context of these various research issues, Section 3 presents related work in the area along with the current state of the project. Section 4 discusses conclusions and future directions for this research effort.

2 The NIST Design Repository Project: Objectives and Research Issues

The NIST Design Repository Project is an ongoing multi-year project with the objective of developing a framework to enable the creation of design repositories. In contrast to traditional design databases, design repositories serve not only as archives, but as repositories of heterogeneous knowledge and data that are designed to support representation, capture, sharing, and reuse of corporate design knowledge.

This research is driven by industry needs that were identified at a workshop held at NIST in November, 1996 [1]. The intent of the project is to impact industry in two ways: (1) by developing basic technological infrastructure to support the creation and use of design repositories in industry, and (2) by providing industry with access to design information that companies can use in their own design processes in order to demonstrate the value of these repositories before significant effort is invested in their development. Thus, in addition to performing basic research in the area of design repositories, the scope of this project also includes the development of sets of domain-specific prototype repositories that will be made available online via the World Wide Web.

Through the course of this project, a variety of research issues have arisen that will in the long term affect the way in which design repositories are implemented and used. These issues include[1]:

- Development of an information modeling framework to support modeling of engineering artifacts, that provides a more comprehensive knowledge representation than traditional CAD systems.
- The use of standard representations, when possible, to leverage research efforts and maximize interoperability with existing software used in industry, and contribution to long-term standards development where standards currently do not exist. The latter consists of the development of information models for representation of knowledge that is generally not found in traditional CAD systems, such as function, behavior, etc.
- A need for representations that are both human interpretable and machine interpretable so that information stored in a repository is accessible and usable by both human designers and knowledge-based design systems that might be used for (partially) automating part of a design process.
- Implementation of interfaces for creating, editing, and browsing design repositories that are easy to use and effective in conveying information that is desired.
- Development of taxonomies of standardized terminology to help provide consistency in, and across, design repositories, as well as to facilitate indexing, search, and retrieval of information from them.

Because of limitations on project resources, the degree to which these issues have been considered in the NIST Design Repository Project varies. However, these issues are all important within our view of the role of design repositories in industry, and ultimately all will have to be addressed by the research community before design repositories can successfully transition into engineering industry.

The goal of this research is not to provide commercial-quality software for creating design repositories, but to provide prototypes that show the utility of the design repository concept. By highlighting and starting to address these issues, this work will speed up the transition and adoption of this kind of technology into industry. The following section outlines the current state of the project and elaborates on each of the above issues within that context. The existing implementation is also described.

---

[1] In addition to these research issues, technological issues such as database and Internet access issues also impact this project. Other pragmatic issues such as security of communications and protection of intellectual property when sharing or exchanging design knowledge will also have to be resolved before design repositories become widely used. Such issues are not the focus of this paper.

3 Current State of the Project

3.1 Information Modeling Framework

Traditional CAD systems are limited to representation of geometric data and other types of information relating to geometry that may include constraints, parametric information, features, and so on. Because of industry's increasing dependence on other types of knowledge in the design process, new classes of tools to support knowledge-based design, product data management (PDM), and concurrent engineering have begun emerging in the engineering software marketplace.

When contrasted with traditional CAD tools, these new systems are making progress toward the next generation of engineering design support tools. However, these systems have been focusing primarily on database-related issues and do not place a primary emphasis on information models for artifact representation (e.g., [2], [3], [4], [5], and [6]). Furthermore, although these systems can often represent non-geometric knowledge (e.g. information about the design process, manufacturing process, bills of materials, etc.), representation of the *artifact* itself is still generally limited to geometry.

Research in the area of intelligent design systems has taken an approach to artifact representation where modeling is typically divided into three main areas of study: the physical layout of the artifact (form), an indication of the overall effect that the artifact creates (function), and a causal account of the operation of the artifact (behavior). Examples of these areas include the qualitative simulation work in [7], behavioral and functional representation in [8], functional representation in [9] and its successor "SBF models" from projects such as KRITIK [10] and INTERACTIVE KRITIK [11], the YMIR project [12], CONGEN [13], and others.

While there are major differences in the implementations of such models, the top level division into representation of form, behavior and function is a common one. The NIST Design Repository Project utilizes a three-tiered approach to artifact representation which incorporates these three concepts. This project, building on previous design representation research, has resulted in an object-oriented representation format that provides a high level division into *form*, *function*, and *behavior* [13], [14].

This modeling language represents artifacts as sets of *objects*, and *relationships*. *Objects* represent physical entities such as assemblies, subassemblies, and components[2], as well as non-physical concepts such as function and behavior. *Relationships* are used to represent the relationship between sets of objects, including a physical decomposition (of an assembly into subassemblies), a functional decomposition (of a function into subfunctions), and other kinds of relationships between ob-

---

[2] In the remainder of this paper, the term *artifact* is used in a generic sense and may refer to assemblies, subassemblies or individual components.

jects. The overall artifact representation is comprised not only of the collection of the objects that represent physical entities, but also the other objects, relationships, the interconnections between them, as well as various attributes and their values.

As is true in general for object-oriented representations, objects and relationships are instantiated from classes that contain attributes and other information that are transferred via inheritance mechanisms. Thus, *a priori* development of useful generic classes of objects and relationships makes modeling of a specific artifact easier since much of the required attributes are already represented in the class schemata. The artifact modeling language and associated object-oriented representation are described in greater detail in [15].

3.2 Knowledge Representation

In order to leverage research efforts and maximize interoperability with existing software used in industry, this work makes use of standard representations when possible. For representation of geometry, this work uses ISO 10303 which is more commonly known as STEP (Standard for the Exchange of Product Model Data) [16] (specifically AP (application protocol) 203 [17]). As more and more CAD companies incorporate the ability to import and export STEP data, the use of STEP AP 203 as a neutral format will make the NIST design repositories more widely accessible than would a proprietary, CAD system-specific format.

In addition to the STEP AP 203 format, geometry is also maintained in VRML (Virtual Reality Modeling Language) format [18] to facilitate visualization of repositories that may be accessed remotely via the World Wide Web. Although STEP AP 203 viewers are available, they are neither as pervasive nor as well integrated with web browsers as are VRML viewers. VRML is not suitable as a complete replacement for STEP AP 203; however, because it was designed for graphical display of geometry and not as a general geometric representation, VRML is not able to represent all kinds of geometric information needed by designers.

Within the standards development community, the focus to date has been primarily on representation of geometric data. This is clearly essential, but it is no longer sufficient. As industry's reliance on non-geometric information and knowledge-based design increases, so does the need for standards for design knowledge such as function, behavior, and other kinds of non-geometric data. In addition to geometry, the information modeling framework that has been developed within this project presently includes representation of function and behavior [15]; if required, the framework will be extended to include other kinds of non-geometric information in the future. Although formal standards for this kind of information are not yet being explored in the standards community, it is hoped that work done within the National Institute of Standards and Technology will lay a foundation for long-term standards development.

3.3 Interpretability of Knowledge Representations

Design repositories are intended to serve as information-rich stores of corporate design knowledge. They are not, however, intended simply to be used as sophisticated part catalogs where a designer can search for final parts or subassemblies that can be dropped into a new design. In most cases, artifact knowledge retrieved from previous designs will not be completely applicable to a similar problem. A design may require further modification before being usable in a new design. In many instances, even a modified design may not be applicable. A designer can still benefit from retrieval of knowledge about previous designs by abstracting information and applying it to a new design, or by gaining insight into how an earlier related artifact was designed. Another way in which stored design knowledge can be used is to support design automation. Knowledge-based design systems are becoming more prevalent in industry because of their ability to automate (or partially automate) design reasoning and in some cases to take on the role of designer for selected portions of the design process.

From the knowledge representation standpoint, these two uses produce conflicting requirements. Although natural language is the most expressive and easily comprehended format for humans, for the purposes of storage and retrieval, as well as computer-based reasoning, natural language is a poor choice and formal representations are preferable. Conversely, a formal representation may be of little use to a human designer if extracting information from it is too difficult. Thus a key objective of this research has been to provide knowledge representations that are both human and machine interpretable, so that information stored in a design repository is accessible and usable by both human designers and knowledge-based design systems. The information modeling framework [15] makes use of a formal knowledge representation, but one that uses data structures that are comprehensible by a human designer, and which lends itself to browsing an artifact representation in a meaningful manner given suitable interfaces.

3.4 Interfaces

The development of extended knowledge representations is necessary but not sufficient in order to successfully use design repositories in industry. Initially, an artifact must be modeled and from this model a design repository must be created. For a design repository to be successfully used for knowledge reuse once it exists, information stored in it must be accessible and easily retrievable by a designer. In many instances, a repository may also be edited and updated through its lifetime. To address these needs, another focus of the current research effort is in the development of interfaces for creating, editing, and browsing design repositories.

There are two ways to create a design repository within the existing implementation: a batch process and an interactive editor. In the batch process, the designer models an artifact by creating an ASCII text file in a specified format that follows the schemata for the data structures in the information modeling framework described previously (objects, relationships, etc.). A compiler process can then be executed from the command line. The compiler parses all the information in this text file and transfers it into a design repository, creating object-oriented data structures and linking them together appropriately. Although the command-line compiler does not have (or need) an interface, in some cases it may be convenient for the designer to have the ability to do artifact modeling “off line” using nothing more than a text editor, and then create a design repository in a batch mode.

The other method for creating a design repository is interactively through a web-based application. This application provides the designer with an interface to the design repository database. Built into this interface are functions for artifact modeling (creating objects and relationships by instantiating them from predefined classes, specifying connections between objects and relationships, filling in attribute values, and so on), as well as some basic consistency maintenance functions.

The other available interface is a web-based artifact browser (see Figure 1). The artifact browser provides a point-and-click interface that allows the designer to browse an existing design repository. The object-oriented data structures (objects, relationships, and their classes) are presented in table form, where hyperlinks allow navigation to connected data structures. From the artifact object shown in the figure, one can click on one hyperlink to examine the artifact’s function, or another to view the relationships for that artifact (one indicating the artifact of which this object is a subassembly, and the other specifying the further decomposition of this artifact into subassemblies and components). Following an artifact’s form hyperlink brings up an object that has a link to a 3D model of that artifact, in either STEP AP 203 format or in VRML format for web-based viewing. The different types of data structures as well as the hyperlinks are color-coded to easily distinguish between them.

In addition to navigating from object to object as described, the bottom right portion of the interface provides the overall hierarchical physical decomposition of an artifact into assemblies, subassemblies, down to individual components. The user can quickly jump to any artifact object in the hierarchy by following one of those hyperlinks. As different parts of the design are visited, they are added to the history in the top right portion of the interface so that the user can quickly return to any object or relationship that was previously viewed during a browsing session.

3.5 Taxonomies and Terminology

Issues of taxonomy and terminology are another important area of focus within this project. One of the main barriers to retrieving and reusing knowledge effectively is problems with terminology: from domain to domain (and sometimes even within a domain) designers use different terms to mean the same thing, or use the same term to mean different things. In order to help provide consistency in and across design repositories, as well as to facilitate indexing, search, and retrieval of information from them, there is a need for standardized terminology and vocabulary. Ideally, this would be done at the research stage of the project and users would have to work

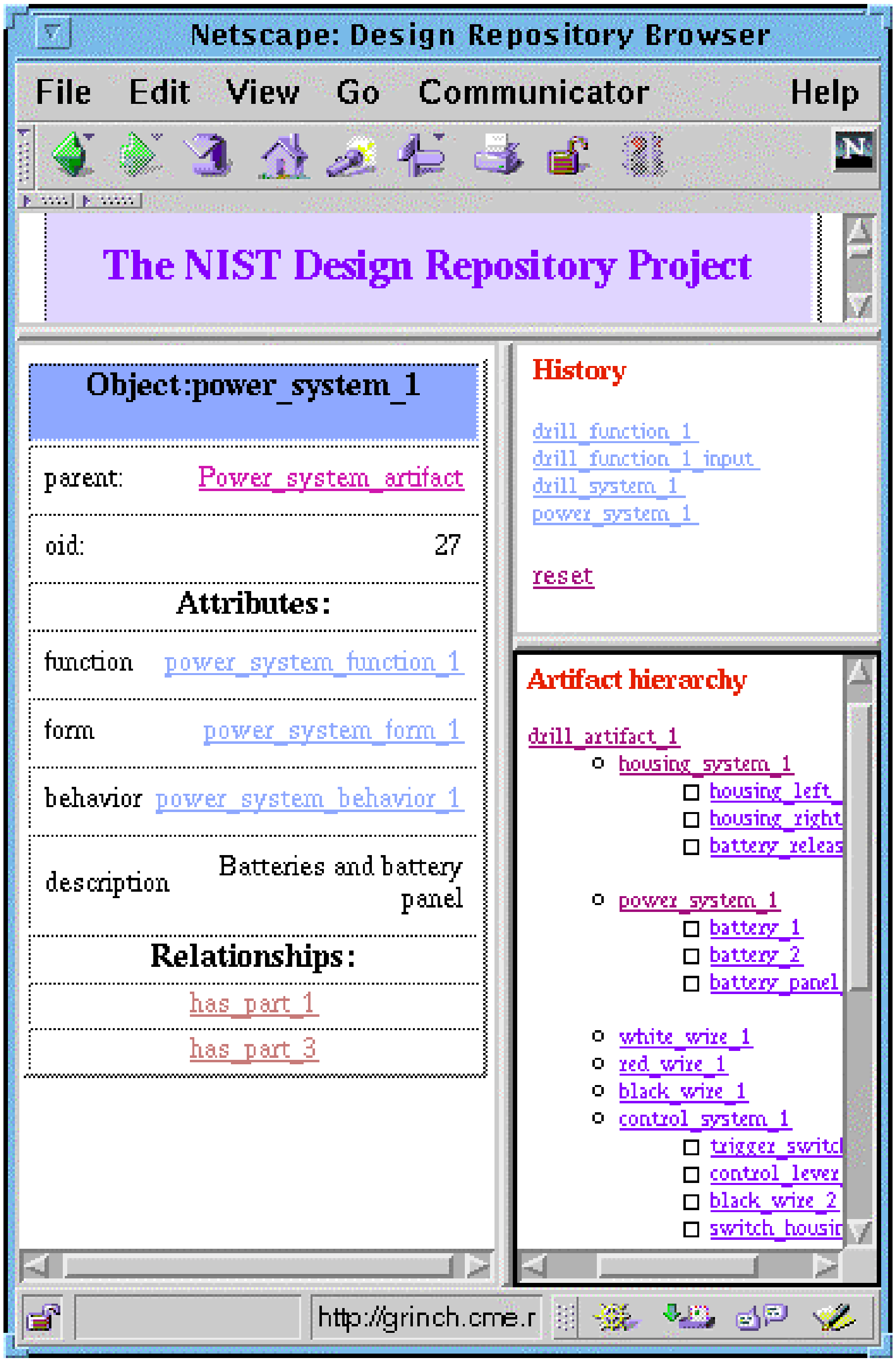

Figure 1. Web-based artifact browser interface.

within the prescribed terminology. Realistically, there are too many conflicts to be resolved independently of domain or application, but allowing a company to develop its own taxonomies will still be beneficial. Extensive work in supporting construction of taxonomies and even more formal ontologies has been done by [19], among others.

A portion of the current research effort is focusing on functional representation, and the development of a comprehensive set of functions and associated flows[3]. The objective is to generate a pair of taxonomies one for engineering functions and one for flows, which is as small as possible, yet generic enough to allow modeling of a broad variety of engineering artifacts. The motivation for creating two separate taxonomies for functions and flows is to avoid a combinatorial explosion in concepts required for modeling artifacts. If flows were considered as part of a function, four different functions would be required to represent the following functions: *transform direct current to linear motion*, *transform direct current to rotational motion*, *transform linear motion to direct current*, *transform rotational motion to direct current*. By considering flows separately, only a single *transform* function, which can have many different flows as input and output, is needed. Related work in this area includes [20], [21], and [22].

Work to date has addressed primarily the representation of function for an artifact and not a mapping of function to behavior, which describes how a function is achieved. Issues associated with behavioral modeling are discussed further in the future work section of the paper.

3.6 Implementation

The current implementation of the NIST Design Repository Project is comprised of a tool suite and a prototype design repository. The tool suite consists of the following components:

- ObjectStore[4], a commercial object-oriented database management system by Object Design, Inc.,
- the design compiler used to create a design repository in batch mode from an ASCII text file in a predefined format,
- a design extractor or “decompiler” that extracts the contents of a design repository and writes that information to a text file in that same predefined format,

---

[3] For example, if an artifact converts direct current to rotational motion, the function is to transform and the associated flows are direct current and rotational motion.

[4] Use of any commercial product or company names in this paper is intended to provide readers with information regarding the implementation of the research described, and does not imply recommendation or endorsement by the National Institute of Standards and Technology.

- STEP/Works, a STEP AP 203 viewer from TechneGroup, Inc., for visualization of three-dimensional geometry,
- the web-based artifact browser and the web-based design repository editor/design modeler, used in conjunction with any of several web browsers,
- any of several VRML viewers or plug-ins, also for visualization of three-dimensional geometry.

The web-based interfaces are client applications that communicate with the database server containing a design repository via common gateway interface (CGI) scripts. The client can be run through most web browsers, allowing remote access of a design repository from any machine on the Internet that can communicate with the machine running the server.

The prototype design repository contains the artifact representation of a Black & Decker cordless power drill. The drill artifact repository contains 28 artifact objects (assemblies, subassemblies, and components) and over 250 additional objects and relationships belonging to 64 (object) classes and 6 relationship classes that describe their form, functions, and behaviors. In addition to representations of form, function, and behavior, the artifact representation includes information about many other attributes (including color, material, and control settings) and parameters (such as battery voltage, output torque, and direction of rotation).

4 Conclusions and Future Directions

The NIST Design Repository Project is working toward the development of an information modeling framework to support the creation of design repositories. This project is driven by industry needs for technology to support the increasing role of knowledge-based design, including the representation, capture, sharing, and reuse of corporate design knowledge. This paper has presented the project objectives, associated research issues, and the current status of the project in the context of those issues. The prototype implementation was also described.

Focus of work to date has been in four areas: (1) the development of a design modeling language for representing design artifact knowledge, (2) the implementation of interfaces for creating and browsing artifact repositories, (3) the creation of a prototype design repository, and (4) work toward identifying a taxonomy of functions and associated flows that is small, yet generic enough to model a broad variety of engineering artifacts. In addition to continuing these aspects of the project and populating the design repositories with new design artifacts, there are several important areas of research that will be addressed by future project milestones.

The functionality of existing interfaces will be extended and new interfaces will be developed. The current artifact browser interface provides the designer with a hierarchical physical decomposition of the artifact. For browsing an artifact from the perspective of function, providing a functional decomposition would also be useful. Additional session management enhancements will also be made, to allow multiple browsing sessions to exist concurrently, to allow a designer to end a session and return to the same state at a later point in time rather than initiating a new session, and allowing definition of multiple users that can set preferences or personalize the interface in various ways. Regarding new interfaces, one important requirement that is not currently addressed is to provide a class editor for creation of new object and relationship classes. The current artifact modeling interface allows the user to create a new repository by instantiating new objects and relationships from predefined classes. However, no set of predefined classes will be universally applicable to all design problems and the ability to extend this set will be important to the designer.

The main aspects of the design modeling language are the representation of form, function, and behavior (as well as relationships between their associated objects). The representation of form (geometry) and function in this project. The representation of behavior within the existing framework is a relatively simple descriptive text-based approach. Another objective for future work is to incorporate more a formal representation of behavior to allow composable simulations. Work in this area is being done as part of two projects that are funded by the Defense Advanced Research Projects Agency (DARPA) Rapid Design Exploration and Optimization (RaDEO) program, for which NIST is a funding agent: the Model-Based Support of Distributed Collaborative Design (previously How Things Work) project at the Stanford University Knowledge Systems Laboratory [23], [24], [25] and the Active Catalog project at the University of Southern California Information Sciences Institute [26], [27]. Discussions are underway to explore the possibility of incorporating an existing behavior modeling language into the NIST Design Repository Project rather than developing an entirely new one.

These attempts at formalizing representations of function and behavior, as well as the other taxonomic issues discussed previously, will facilitate the tasks of indexing design knowledge for efficient search and retrieval. However, aside from formal representations and a standardized vocabulary, the indexing and search mechanisms must themselves be developed. This is an area of long term future research, as much work remains to be done before this problem can be adequately addressed.

Once design repositories are more broadly used by industry, the question of how to exchange information between repositories will become more important. If a standardized design modeling language is developed (possibly, but not necessarily, using the framework developed in this project as a basis), repositories themselves may not be interchangeable between systems. If two design repository development systems are built on top of two different object-oriented database software systems, one will not be able to use a repository created by the other without some kind of translation step. The ASCII text file format used as part of this work is inadequate for that purpose because there is no mechanism to include a computer interpretable description of the knowledge schemata or the file content within the file itself.

The Extensible Markup Language (XML) [28] and Resource Definition Framework (RDF) [29], [30] are being examined as a possible solution to this problem. Though other work has been done in developing formal approaches for the exchange of knowledge (e.g. Knowledge Interchange Format (KIF) [31] and Open Knowledge Base Connectivity (OKBC) [32], [33]), XML and RDF have the main advantage of widespread adoption in the information technology world; XML (and likely RDF) support is expected in upcoming versions of several commercial web browsers and word processing applications. This research direction is being explored in order to provide a more broad-based solution to an industry which is increasingly looking towards purchase of off-the-shelf software over in-house development when possible.

The NIST Design Repository Project was motivated by needs identified at an industry workshop held at NIST in November 1996 [1] and, as such, it is expected that industry will benefit from the technical foundation for the development of design repositories provided by this research. Another potential use for this work is to enhance the content of patent databases residing at the U.S. Patent and Trademark Office. Currently, patent information consists of text and two-dimensional images. Providing a formal artifact representation that extends to function and behavior not only leads to a more comprehensive description of a device, it also provides additional types of information for meaningful indexing, which can lead to more efficient search and retrieval of patent information.

References

[1] Szykman, S., R. D. Sriram, and S. J. Smith (eds.) (1998), *Proceedings of the NIST Design Repository Workshop*, NISTIR 6159, National Institute of Standards and Technology, Gaithersburg, MD, November, 1996.

[2] Hardwick, M. and D. Loffredo (1995), "Using EXPRESS to Implement Concurrent Engineering Databases," *Proceedings of the 1995 ASME Computers in Engineering Conference and Engineering Database Symposium*, Boston, MA, September.

[3] Bliznakov, P. I., J. J. Shah, and S. D. Urban (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.

[4] Kim, T. S., S.-H. Han, and Y. J. Shin (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.

[5] Shah, J. J., D. K. Jeon, S. D. Urban, P. Bliznakov, and M. Rogers (1996), "Database Infrastructure for Supporting Engineering Design Histories," *Computer-Aided Design*, **28**(5).

[6] Wood III, W. H. and A. M. Agogino (1996), "Case-Based Conceptual Design Information Server for Concurrent Engineering," *Computer-Aided Design*, **28**(5).

[7] de Kleer, J. and J. S. Brown (1983), "Assumptions and Ambiguities in Mechanistic Mental Models," *Mental Models*, D. Gentner and A. L. Stevens (eds.), Lawrence Erlbaum Associates.

[8] Iwasaki, Y. and B. Chandrasekaran (1992), "Design Verification through Function and Behavior-Oriented Representations: Bridging the Gap between Function and

Behavior," *Artificial Intelligence in Design '92*, J. S. Gero (ed.), Kluwer Academic Publishers, Boston.

[9] Chandrasekaran, B., A. Goel, and Y. Iwasaki (1993), "Functional Representation as Design Rationale" *IEEE Computer*, January, pp. 48-56.

[10] Goel, A., S. Bhatta, and E. Stroulia (1996), "Kritik: An Early Case-Based Design System," To appear in *Issues in Case-Based Design*, M. Maher and P. Pu (eds.), Hillsdale, NJ, Erlbaum.

[11] Goel, A., A. Gomez, N. Grue, J. W. Murdock, M. Recker, and T. Govindaraj (1996), "Explanatory Interface in Interactive Design Environments," *Artificial Intelligence in Design '96*, J. S. Gero (ed.), Kluwer Academic Publishers, Boston.

[12] Alberts, L. K. and F. Dikker (1994), "Integrating Standards and Synthesis Knowledge Using the YMIR Ontology," *Artificial Intelligence in Design '94*, J. S. Gero (ed.), Kluwer Academic Publishers, Boston.

[13] Gorti, S. R., A. Gupta, G. J. Kim, R. D. Sriram, and A. Wong (1998), "An Object-Oriented Representation for Product and Design Process", To appear in *Computer-Aided Design*.

[14] Wong, A. and D. Sriram (1993), "SHARED: An Information Model for Cooperative Product Development," *Research in Engineering Design*, **5**(1):21-39. Extended version: *Shared Workspaces for Computer-Aided Collaborative Engineering*, MIT Intelligent Engineering Systems Laboratory, Document 93-06 (technical report), Massachusetts Institute of Technology, Cambridge, MA.

[15] Murdock, J. W., S. Szykman, and R. D. Sriram (1997), "An Information Modeling Framework to Support Design Databases and Repositories," *Proceedings of the 1997 ASME Design for Manufacturing Conference*, Paper No. DETC97/DFM-4373, Sacramento, California, September.

[16] ISO 10303, *Industrial Automation Systems and Integration – Product Data Representation and Exchange*, ISO TC184, SC4 (ISO Technical Committee 184, Subcommittee 4).

[17] ISO 10303-203:1994, *Industrial Automation Systems and Integration – Product Data Representation and Exchange -- Part 203: Application Protocol: Configuration Controlled Design*, ISO TC184, SC4 (ISO Technical Committee 184, Subcommittee 4).

[18] ISO/IEC 14772, *Information technology – Computer graphics and Image Processing - The Virtual Reality Modeling Language*, ISO/IEC JTC1, SC24 (ISO/IEC, Joint Technical Committee 1, Subcommittee 24).

[19] Farquhar, A., R. Fikes, and J. Rice (1996), "The Ontolingua Server: a Tool for Collaborative Ontology Construction," *Proceedings of the Knowledge Acquisition Workshop*, B. Gaines (ed.), Banff, Canada.

[20] Kirschman, C. F., G. M. Fadel, and C. C. Jara-Almonte (1996), "Classifying Functions for Mechanical Design," *Proceedings of the 8th ASME International Conference on Design Theory and Methodology*, Paper No. 96-DETC/DTM-1504, Irvine, California, August.

[21] Little, A., K. L. Wood, and D. McAdams (1997), "Functional Analysis: A Fundamental Empirical Study for Reverse Engineering, Benchmarking and Redesign,"

Proceedings of the 9th ASME International Conference on Design Theory and Methodology, Paper No. DETC97/DTM-3879, Sacramento, CA, September.

[22] Stone, R. B., K. L. Wood, and R. H. Crawford (1998), "A Heuristic Method to Identify Modules from a Functional Description of a Product," *Proceedings of the 10th ASME International Conference on Design Theory and Methodology*, Paper No. DETC98/DTM-5642, Atlanta, GA, September.

[23] Falkenhainer, B., A. Farquhar, D. Bobrow et al. (1994), *CML: A compositional modeling language*, KSL-94-16 (technical report). Stanford Knowledge Systems Laboratory, Stanford University, Stanford, CA.

[24] Iwasaki, Y., A. Farquhar, R. Fikes, and J. Rice (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.

[25] Chouiery, B. Y., S. McIlraith, Y. Iwasaki et al. (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; revised version to appear in *Proceedings of the 12th International Workshop on Qualitative Reasoning (QR '98)*.

[26] Ling, S. R, J. Kim, P. Will, and P. Luo (1997), "Active Catalog: Searching and Using Catalog Information in Internet-Based Design," *Proceedings of the 1997 ASME Computers in Engineering Conference*, Paper No. DETC97/CIE-4292, Sacramento, California, September.

[27] Luo, P. and P. Will (1997), "Active Catalog: A Knowledge-Rich Design Library Facilitating Information Consumption," *Knowledge Intensive CAD, Volume 2 - Proceedings of the IFIP TC5 WG5.2 International Conference on Knowledge Intensive CAD*, Mäntylä, M., S. Finger, and T. Tomiyama, (eds.), Pittsburgh, PA, September, 1996.

[28] World Wide Web Consortium (1998), *Extensible Markup Language (XML) 1.0*, World Wide Web Consortium (W3C) Recommendation, February.

[29] Lassila, O. and R. R. Swick (1998), *Resource Description Framework (RDF) Model and Syntax Specification*, World Wide Web Consortium (W3C) Working Draft, February.

[30] Brickley, D., R. V. Guha, and A. Layman (1998), *Resource Description Framework (RDF) Schemas*, World Wide Web Consortium (W3C) Working Draft, April.

[31] Genesereth, M. R. and R. E. Fikes (1992), *Knowledge Interchange Format, Version 3.0 Reference Manual*, KSL-92-86 (technical report), Stanford Knowledge Systems Laboratory, Stanford University, Stanford, CA.

[32] V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Karp, and J. P. Rice (1998), "OKBC: A Programmatic Foundation for Knowledge Base Interoperability," To appear in: *Proceedings of the 15th National Conference on Artificial Intelligence – AAAI-98*, Madison, WI.

[33] V. K. Chaudhri, A. Farquhar, R. Fikes, P. D. Karp, and J. P. Rice (1998), *Open Knowledge Base Connectivity 2.0*, KSL-98-06 (technical report), Stanford Knowledge Systems Laboratory, Stanford University, Stanford, CA.