A Case-Based Conceptual Design Information Server for Concurrent Engineering[1]

William H. Wood III and Alice M. Agogino
Dept. of Mechanical Engineering, U.C. Berkeley

Abstract

Conceptual design requires processing information from diverse sources in order to define the functional requirements, operating constraints, and evaluation criteria pertinent to accomplishing a prescribed goal. This goal has traditionally been focused on the functionality of an artifact to be provided to a customer. Concurrent engineering redefines this focus to account for the concerns of 'customers' not previously considered - those along the entire life cycle of an artifact: marketing, design, manufacture, distribution, operation, and disposal. Expanding the design focus to include all of these customers places far greater informational demands on the designer. Case-based reasoning applies experience stored in a computerized form toward solving similar problems in slightly altered contexts. It has been applied successfully to routine design where both the form and the content of design information can be encoded symbolically and manipulated using artificial intelligence techniques. Concurrent conceptual design presents unique problems for such an approach because design information must be considered at many levels of abstraction and from many viewpoints. We describe an approach to providing the designer with case-based conceptual design information stored in the richly expressive medium of hypermedia (hypertext incorporating multimedia). Design cases of industry best practices in concurrent engineering are indexed to provide access through multiple interfaces, allowing the user to browse, explore, or pinpoint design case information. The conceptual design information server (CDIS) is implemented using emerging internet standards, such as those associated with the World Wide Web (WWW) and Wide Area Information Service (WAIS), coupled to a robust Structured Query Language (SQL) database and traditional CAD packages.

1. Introduction

Studies by the the National Research Council (1991) and others (Institute for Defense Analysis, 1988) of large scale projects estimate that 80-90% of the life cycle design costs (including fabrication, construction, energy, maintenance and disposal) are determined in the first 10-20% of the design phase - conceptual design. Drawing on a study of a major automobile manufacturer, Salzberg and Watkins(1991) conclude that lack of information in the conceptual design phase is the major cause for most of the upstream life cycle design problems.

Designers draw on a broad range of experience to carry out conceptual design and it is this experience level and the associcated information that distinguishes the novice designer from the expert. Experienced designers apply their extensive knowledge base to design problems, consciously or unconsciously using requirements, constraints, and criteria from past designs in decomposing the problem and evaluating alternatives. Unfortunately, the design complexities and range of disciplines associated with concurrent engineering may require knowledge beyond the direct experience of most practicing engineers. All designers are novices in some contexts. A natural means of augmenting designer experience with insights gained from stored design instances is the goal of case-based approaches to design.

[1] Submitted for review to *Computer Aided Design*.

Providing access to successful cases of concurrent design is the focus of the Conceptual Design Information Server (CDIS) described in this paper. Modular case studies in which life cycle issues and initial customer specifications drive conceptual design toward establishing functional requirements, operating constraints, evaluation criteria, and an initial design are documented, indexed and stored. CDIS is designed to support concurrent engineering activities throughout conceptual design, providing access to stored experience at varying levels of representational abstraction. Context relevant life cycle information is presented to the designer as appropriate to enhance the effectiveness and efficiency of the resulting design.

A national database called NEEDS, the National Engineering Education Delivery System (Agogino et al., 1993) has been developed within the Synthesis Coalition[2] to make these case studies and

[2] The *Synthesis* Coalition, supported by the National Science Foundation and industrial partners, is comprised of the following eight educational institutions: California Polytechnic State University at San Luis Obispo, Cornell, Hampton, Iowa State, Southern, Stanford, and Tuskegee Universities, and the University of California at Berkeley.

other engineering courseware available to educators, students, and practicing engineers. CDIS extends this architecture with interactive information retrieval interfaces that support conceptual design through indexing at various levels of granularity matching the level of abstraction to the immediate need in conceptual design. These interfaces are built on international network standards to create a distributed networked conceptual design information server usable across personal, group, corporate, national, or international boundaries.

Section 2 of the paper casts traditional case-based reasoning techniques from mechanical, structural, and architectural design into the framework of conceptual design, suggesting a shift in the case-based reasoning paradigm toward supporting large, hardly decomposable problems. Section 3 describes the current case base and experience using it in an educational environment. The system architecture developed to support conceptual design through case-based reasoning is described in Section 4 and its implementation is illustrated in Section 5. Finally Section 6 provides a road map to future development of both case base and information service within CDIS.

2. Case-Based Reasoning in Design

Case-based reasoning has been an obvious paradigm for application of AI techniques within computer-aided design. Expert designers draw heavily from case experience, modifying former results to fit new contexts or objectives. To the extent that this is the design paradigm needed to solve a problem, case-based reasoning as described in Figure 1 has been successful. Examples of successful applications domains are: extrusion die design (Jamalabad and Langrana, 1993), aerospace autoclave loading (Henessey and Hinkle, 1992), and aerospace weld process specification (O'Conner et al., 1992). In each of these domains, theoretical analysis is expensive, goals are straightforward, and empirical data can fully describe the cases. With CADET, Navinchandra(1988) adds a restricted domain theory to case-based reasoning in fluid system design; goals are represented by behavior and cases are built from the synthesis of artifacts represented in the same behavioral language. Howe et al.(1986) describes DOMINIC, a multi-domain parametric design system in which more general design goals are explicitly represented related to design parameters through heuristic measures. DOMINIC has since been extended by Dixon et al. (1992) with expansion of representational schemes toward encoding more diverse mechanical design domains, especially those of interest within concurrent engineering. Solutions are, however, synthesized within the various contexts of these different representations; no canonical set of design features has yet been identified to unify the knowledge. Conceptual design within DOMINIC builds on previous design cases through a mapping from phenomenological representations like those used in CADET to system embodiment in the form of stored artifact behavior (Dixon et al., 1993).

Structural design has been an equally rich application domain for case-based reasoning. Zhao and Maher (1988) used an analogical reasoning foundation in STRUPLE. Maher and Zhang (1991,1993) describe CADSYN, an architecture based on case decomposition and transformation knowledge to structure cases for reuse. CADSYN employs a symbol-based representational language; specifications and data are stored homogeneously through attribute value pairs. A rule-based constraint satisfaction approach transforms cases toward application in new contexts. Howard (1991) separates the knowledge sources within structural design into general and project specific classes to identify the aspects of past cases that are valid for reuse. Garcia and Howard (1992) extend this idea into a system where decision justification during design is used to acquire domain specific (constraints) and project specific (rationale) knowledge through an intelligent design documentation system.

Extending the design of buildings to include architectural goals and specifications has stressed the representational schemes that have been successful in structural design. Hua and Faltings (1993) present issues in CADRE toward implementing an architectural design system which includes reasoning about topologies and spatial relationships. In this work, they conclude that the many levels of abstraction of both specification and case information provides a formidable challenge to the case-based reasoning paradigm. Pearce et al. (1992) recount experience in developing ARCHIE, a case-based architectural design tool. Again, the case-based reasoning paradigm proved difficult to implement successfully due to the difficulty of representing large cases and problems in modeling specifications and preferences and their successful implementation within a case. These difficulties are significant in that architectural design in these contexts *is* conceptual design.

In concurrent conceptual design, as in architectural design, the size of a design case increases compared to routine design because of the multiple interrelated objectives and constraints generated by the multidisciplinary approach of concurrent engineering and the multiple customer approach of life cycle design. A case-based architectural design system, ARCHIE, has yielded the initial conclusion that such design cases are quite large and difficult to represent in a single language that can translate among different design contexts using typical symbol-level case based reasoning methods. Domeshek and Kolodner (1993) make the following suggestions toward effectively representing and applying case-based reasoning to conceptual design:

1. Organize cases into short, pointed presentations that teach specific lessons based on particular experiences.
2. Index such stories in terms of design situations they address.
3. Describe design situations in terms of design issues associated with particular structural or functional parts of an artifact, and remember to consider issues arising from all phases of the artifact's life-cycle from the points of view of all relevant stakeholders.
4. Explicitly note interaction between design issues to broaden the user's focus and draw their attention to related aspects of a design with which they should be concerned.
5. Link stories of specific successes and failures to general guidelines which in turn link back to other related stories in order to allow the user to easily explore a range of responses to the same basic issue.

These recommendations suggest a paradigm shift in conceptual design environments away from the typical symbolic AI case-based reasoning methodology shown in Figure 1 toward one supporting directed exploration through the design case base, shown in Figure 2. Issues within the implementation remain similar: indexing the case base for effective retrieval and enabling the integration of information from past designs toward the solution of new design instances. However, emphasis is now shifted from automatic design to automated design support through increased access to relevant information.

3. The Case Base

Agogino and Hsi (1993a) describe support for exploratory learning in engineering design through a set of multimedia case studies. These case studies have been developed for educational purposes within the *Synthesis* NSF Engineering Education Coalition in a collaborative effort with industry partners (Agogino et al., 1992). Each brings a domain-specific view of life cycle design, multidisciplinary engineering problem solving, and concurrent engineering solutions to the user. Links to analysis, history, simulation, CAD drawings and experimentation are included where appropriate. On-line engineering texts augment the case studies, providing theoretic background underlying the design decisions presented in the case study. Other links tie the design activity to specific artifacts to help ground the reader in the context of the design project. All of this information is organized around a product life-cycle navigational backbone, or concept map. Design cases represented in this set include: the IBM Proprinter, Mattel toy design and manufacture, Saturn automobiles, the Ingersoll-Rand Cyclone Grinder, disk drives, and the

U.C. Berkeley Human Powered Vehicle project.

The content of these cases and the relationships among them make for a particularly strong base of knowledge for concurrent life cycle design. In each case, the product realization process described represents industry 'best practices', often a conscious departure from former corporate design practice toward concurrent engineering principles. The Proprinter is a landmark product for IBM, restructuring design practice within the corporation to enable the manufacture of low cost, high volume, consumer oriented products. Toys highlighted in the Mattel case study represent efforts at integrating concurrent engineering into a design process where time-to-market and low cost are the overriding considerations. Saturn is General Motors' test case for the profitability of concurrent life cycle design in the automobile manufacturing arena. For Ingersoll-Rand, the Cyclone grinder demonstrates the value of integrating industrial design's focus on customer recognition and concern into products which had traditionally been engineered for function rather than designed for customers. The Human Powered Vehicle case study is one which demonstrates small scale custom manufacturing, integrating testing and emphasizing evolution in a design which has set world records for performance. The disk drive case is one in which a highly competitive international market is driven by manufacturing quality, mechatronic design, and precision engineering in order to provide the highest storage density and thus best price-performance ratio.

Collectively the case base spans a wide range of design topics by following each product from the identification of marketing opportunity through the conceptual design and then the manufacturing process. Because in each case concurrent design was an important contributor to the overall project and product success, connections among these activities are naturally stressed. Taken as a single presentation, each case illustrates an instance in which a subset of concurrent design practice was applied; as a group, they present concurrent design as practiced over a wide range of product cost, design complexity, manufacturing philosophy, and enterprise organization. Specific topics addressed include: design for assembly, design for automated assembly, flexible assembly process, manufacturing process impact on quality, ergonomic design, ergonomic manufacturing, design for serviceability, and many others. Details on how some of these ideas were applied within a design case and presented for reuse are illustrated through a discussion of the IBM Proprinter case.

The IBM Proprinter case study chronicles the design and manufacture of the best selling dot matrix printer in the world through a series of multimedia presentations incorporating textual, graphic, and video information formats. The presentations are organized around a central navigational backbone, or concept map, which follows the life cycle of the product, from the identification of the need to sell a printer for use with IBM's best-selling PC line, through the decision process leading to the choice of manufacturing the printer in-house and in the United States, through the design for manufacture principles necessary to make a profit in such an undertaking. Manufacturability, specifically design for automated assembly, is the main thrust of the case study just as it was in the design process itself. Interviews with the designers explain the decision process and methods used to reduce part count and eliminate screws, belts, and coil springs from the Proprinter. Video of assembly sequences ably demonstrate the results of their efforts. The design of new manufacturing processes, like that used to injection mold a plastic lead screw used to replace the printer head drive belt, and innovative application of well known techniques from other product types, like the study of toys for designing snap fits, inspires exploration and breadth in developing conceptual designs using concurrent engineering principles. Lessons learned throughout the process include the abandonment of a several mile long flexible automated assembly line for production for assembly of the same design on a compact manual assembly line. This experience relates design for automated assembly to design for assembly in a very convincing fashion, motivating the use of concurrent design techniques to a wide range of manufacturing environments.

A typical presentation of a design issue taken from the IBM Proprinter case study is shown in Figure 3. The textual information describes the advantages of using complex plastic parts to replace the multi-part metal brackets used to locate the paper feed system within the printer. While the metal parts were cheap to manufacture, their replacement reduced the part count in the printer by about 40. In particular, springs and fasteners were molded into these side rails exploiting the idea of function sharing within a component. Design concerns like bearing surfaces for rotating shafts and electrostatic discharge which were not significant in their metal counterparts proved problematic in the plastic design. These issues were resolved partially by reinforcing bearing surfaces with molded-in inserts and adding electrically conductive carbon fiber to the plastic as it goes in to the mold. Other presentations associated with the frame rail design include the layering of parts for easy assembly, the design of the snap fit fastening system, and the design of the molded in spring used to press paper against a moving pinch roller. The navigation buttons at the bottom of Figure 3 can be used to take the user to issues of similar context to this presentation along with menus of interest within the case and links to other cases in the case base (additional extensions are being made to provide access to other design support tools sharing the Mosaic supported protocols like catalogs (Bradley et al., 1994), analysis, and CAD packages). In this way, each case within the case base provides a local structure for browsing; the CDIS provides a means of browsing over many cases, merging conceptual ideas and issues across corporate, industry, or international boundaries.

Testing of the individual case studies has proven them to be very effective in communicating not only the technical information presented in the case but also the rich contextual information that is vital toward supporting open-ended design problem solving (Agogino and Hsi, 1993b, Carlstrom, 1993). These studies are in line with similar results of design studies of 'learning to do' which compare hypertext to non-hypertext reference sources (Lehto and Zhu, 1992). Petroski (1993) proposes case studies of design, specifically classic design failures, as a primary means of communicating design methodology to engineers. Representing conceptual design cases as multimedia presentations lends the communication power of hypertext and rich presentation modes to provide context to information that is difficult to represent in a structured way. A framework of linked multimedia objects based on these multimedia case studies project is the initial case base within CDIS.

4. System Architecture

CDIS is structured around the information flow in conceptual design. This information begins with the needs of the customer. Hauser and Clausing(1988), with their 'House of Quality' QFD method, map subjective customer-oriented design criteria onto mathematical metrics which can objectively evaluate alternatives. With ELK, Hauge and Stauffer (1993) provide a systematic methodology for testing the validity of such mappings between customer's desires and design requirements. If the objective measures are stored without the original language of customer intent, they cannot effectively be contextualized. This goal-to-objective function translation is a primary task within conceptual design where the mapping is the iterative process of defining requirements, constraints, and criteria for evaluation. The information driving such subjective to objective translations must be preserved and indexed to provide contextual information, linking loose design requirements to mathematical evaluation models to fully capture the rationale behind design decisions. Because this tends to encode the process of doing conceptual design, documents representing such information are labeled process-centered in Figure 2.

One of the major challenges in concpetual design is operating on information at several levels of abstraction simultaneously. A natural progression takes place in which requirements, constraints, and criteria parameterize the desired functionality. The design process becomes one of decomposition, imposing boundaries on subsystems so that they may be considered outside of the whole of the conceptual design. Eventually function is mapped onto artifact; this mapping is not necessarily one-to-one for good design takes advantage of function sharing within an artifact. This means that decomposition cannot be decoupling; each element of a design must provide information toward the development of other subsystems. This information must be provided through a set of context specific 'views' reflecting the depth of knowledge needed for the design of other system element. Empirical research bears this progression out: designers consider artifacts at multiple levels of abstraction simultaneously during conceptual design (Stauffer and Ullman, 1988, Stauffer et al., 1987). Ullman (1993) describes the changes in abstraction level throughout the course of design; representation of specification, function, and behavior becomes more precise as the design evolves. Information access must reflect the changing needs of the designer, providing indices on both abstract and concrete terms.

In addition to supporting index interfaces that allow for multiple abstraction levels over the same information, it is important to support the design activities that reflect the varying level of abstraction within the design. Industrial designers browse through magazines identifying forms within products that suggest various functional and interface information. It is more difficult for an engineer to do this when the information is of a dynamic nature; digital video provides a more communicative interface to functional information than does a still image. So an interface to browsing information in various media formats is necessary. As browsing becomes directed toward specific concepts and data elements, it is important that they be placed in context within the case in which they appear. Links from individual elements into presentations supplying this contextual information are provided in CDIS. Because these presentations are themselves part of a larger case study, navigation from a starting point derived from unstructured browsing provides even more context to the concept under consideration. Design information like interviews with the design team, video of manufacturing processes, component descriptions and models, or analysis results are modular data elements in CDIS stored in a standard SQL (Structured Query Language) relational database reflecting the structure of the underlying multimedia case base. This database is indexed using a hierarchical set of keywords offering various abstraction levels over process-centered, function-centered, and artifact-centered information.

Within CDIS, design documentation is also directly content searchable through an index on the text used in a document (and textual descriptions of images and videos) through a Wide Area Information Server (WAIS) system. Although the WAIS natural language indexing scheme can theoretically identify documents of interest in response to unstructured queries, the document content itself is based on the descriptive language used in the case study. Thus the index is based on a language which might be shared within a design group but might not cross discipline or corporate boundaries, hampering ultimate effectiveness. One solution to this interpretation problem might be to prescribe a language for design. While this works for well specified problems as proven by the success of traditional case-based reasoning in such domains, it would not suit the evolutionary and uncertain nature of design information during conceptual design. Instead, it is necessary is to establish an index on *meaning*, not *expression*. The natural language representation used in the case base is the closest to an unbiased representation available, able to relate both information and the its degree of relevance - knowledge. Meaning, in this context, can be sufficiently represented within an intelligent thesaurus used to augment queries to the content-based index. This thesaurus is generated from the same set of keyword used to index information within the case base. The standard free-text query interface indexes into the same thesaurus that provides meaning to the structured query interface. For example a free text query entered by the user: 'what are issues in design for manufacture' would automatically entail subterms like 'design for assembly' and 'design for quality' as well as related terms like 'DFM', 'DFA', 'assembly', etc. In this way, content that is *about* design for maunfacture but does not include that actual text will be retrieved.

Structure within each case study stored in CDIS design document web is provided by a hypertext index, reachable from every document in the design case. This index uses a standard format to link to compound documents containing a design's specifications, subsystem decomposition, and artifacts. This provides a common, omnipresent interface to design documentation without restricting the content or format of the documents themselves. It also provides an entry point to information about design modeling, constraint, and objective functions. Design case storage can be organized to promote readability and enhance life cycle issue communication. Structure is imposed on the design documents after they have been created; designs are not shoe-horned into a predefined representation, the representation structure is adapted to meet the needs of the design case. An additional benefit of such an index is the security it provides users who might get lost in the web of information and need a 'home base' to show where they have been and where they might like to go (Hsi and Agogino, 1993b).

5. System Implementation

The implementation of the case-based CDIS applies network standard interfaces to the basic tasks of conceptual design as described above. The choice of separate interfaces for structured, semi-structured, and unstructured queries provides access to the case base on the abstraction level appropriate to the stage of conceptual design in which the user is involved.

Figure 4 shows the overall architecture of CDIS. Browsing is accomplished through either of two network standard interfaces to the same SQL database of case study components: an X-Window client ImageQuery (Morgan and Jacobsen, 1989) or one implemented in NCSA Mosaic and hypertext transfer protocol (HTTP). Each addresses the SQL database to form queries to pinpoint information for visual browsing based on information type, format, and content. Whereas ImageQuery is able to generate and process arbitrarily complex SQL queries, the Mosaic interface is restricted to a general subset of queries due to the medium in which it is implemented. This tradeoff in query effectiveness is balanced by the ability of the Mosaic interface to provide navigation from the visual browser directly to the case study, providing quick access to contextual information. This is because the interface is implemented in the same network standard hypermedia language as the design cases themselves: hypertext mark-up language (HTML). In addition, HTML and HTTP combine to provide access to CDIS content based WAIS index and to resources available throughout the World Wide Web (WWW). Thus access is gained by navigation, content index, or the ontological index available through the SQL database. Structure is available where appropriate, unstructured access is also supported. The specifics of the system architecture are described below.

<u>Case Study Archives</u>

The SQL archive database at the core of CDIS includes paths to files that store the case study including: design specifications, functional decompositions, artifacts, objectives, models, browse images (thumbnails), digitized video, etc. Each design case is stored as a series of active, hyperlinked WWW browser HTML files which link to other HTML documents or supporting information stored in native format (e.g. analysis program input or CAD files). The entire system is accessed through the network, minimizing demands on the local computer system. Because network standard, interoperable interfaces have been chosen for the implementation, the system can also be easily distributed. Browsing or querying the case-base can pinpoint useful information and download it for further evaluation, reducing the resource commitment required of the server. Design case navigation is the primary mode of exploration in the system. It is augmented by hierarchical ontological indexing in the SQL database and content based indexing. All of these features are set in two primary internet information services: World-Wide Web (WWW) and WAIS (Wide Area Information Service).

<u>WWW</u>

World Wide Web, an internet project begun at CERN with the goal of distributing hypertext information over computer networks, provides CDIS with standard file formats for creating hypertext documents - HTML (hypertext markup language, a document type description of SGML ISO 8879) and a standard protocol for serving and browsing these files - http (hypertext transfer protocol). Textual information within HTML is implemented as ASCII characters with special escape 'tags' that are interpreted by the client for formatting information. Hypertext in HTML is implemented as special 'anchors' that point to resource links to other documents from http servers throughout the network. Hypermedia is implemented through special file type declarations that map to external applications (generally multiplatform standard and freeware oriented). Full hypermedia documents can be served through http, with limitations on document formatting and compromises on interactivity in response to the need for network access simplicity. WWW also includes links to Gopher and WAIS as well as ftp and remote procedure scripts. Figure 3 is an example of the case study implemented in HTML displayed by Mosaic, Figure 5 shows the SQL query interface to CDIS.

WAIS

WAIS, or Wide Area Information Service, is a means for serving information through a robust, Z39.50 compliant protocol (Kahle et al., 1992). The fundamental goal of the WAIS architecture is to provide a large degree of access to data without encumbering the user or the server with knowledge of the underlying document structure or location. WAIS servers provide access to textual documents through a large, inverted index on the actual document text (in the case of non-textual documents, the file name is indexed). As such, it provides a perfect indexing tool for HTML documents (it can be 'taught' to ignore tags and hypertext 'anchors' quite readily). WAIS has also attracted great interest from technical publishers, including our industrial partner John Wiley & Sons. This has afforded us the opportunity of indexing commercial material and linking it into our hypertext documents. Thus when a user wants to browse theoretical background in the context of a design case, it is available through a 'connection' to a publisher database. WAIS servers are accessible through WWW: Figure 6 shows a typical session through a Mosaic interface.

The various technologies identified above are meshed to form an extremely powerful hypermedia information server in support of a case-based architecture for integrated design as shown in Figure 4. The system provides simultaneous support for structured and unstructured queries on both the knowledge and symbol levels. Browsing can be done along levels of granularity from whole design cases to functional decompositions to artifacts to supporting documents. In addition, new modules developed in HTML (which can include links existing remote elements simply by referencing them), need not be downloaded to a local computer at all. Evolving designs can be created and automatically have hyperlinks updated when appropriate. Design cases are also automatically content indexed (at least on textual information) so that arbitrary, query-oriented hyperlinks can be made without being preprogrammed. In this way, 'connections' hard coded in the original design case documents become more fluid and encourage jumps to the most relevant and up to date information available, a significant enhancement to multimedia case study functionality.

6. Conclusions and Future Directions

A new formulation of the case-based reasoning paradigm has been presented for the domain of conceptual design. Emphasis has shifted away from completely automating conceptual design solution toward more effectively communicating design information to the designer. Multiple interfaces support typical conceptual design and case-based reasoning activities such as: context sensitive case retrieval, concept query processing, and multiple abstraction level browsing. The varying degree of design information and commitment typical of an evolving solution is supported through content based indexing on design cases represented by both multimedia, natural language compound documents and structured, relational database encoded ontology.

Much of the reasoning involved in the system has been transformed from case retrieval and transformation to information collection and annotation by the user. This new paradigm is supported by a shift in representation from strong domain bias to expressive, universal languages. Multiple, interconnected representations are used to address the hardly decomposable nature of life cycle conceptual design. Normal case-based activities are performed by the user, gathering and reusing case information at appropriate abstraction levels throughout the conceptual design process.

The CDIS architecture is being tested as part of a database of courseware material in design and synthesis being developed by the Synthesis Coalition - called NEEDS, the National Engineering Education Delivery System. Interested readers are welcome to login into the test bed by browsing the WWW server: http://pawn.berkeley.edu/CDIS.html or requesting a manual by sending email to: aagogino@euler.berkeley.edu.

The architecture provided to accomplish these goals is an open one based on robust, network standard interfaces and protocols. This foundation promises to provide wide access to the present system and future compatibility to network-based commercial information enterprises like design databases or virtual corporations. The system is presently being extended to support mechatronic design; the multidisciplinary mechanical, electrical, and computer software aspects of designs in this domain are ideally matched to the general purpose, domain independent, case-based paradigm established for CDIS, the Conceptual Design Information Server.

Future directions include entension of the case base toward more active design documents based on the same structure used in the current case base. This structure promotes the effective presentation of design cases in hypermedia form; the DesignSCRIBE (Bradley and Agogino, 1991c) and SHARE (Toye et al., 1993) projects provide foundations for encoding design decomposition (function-centered representation) through document templates with decomposition links to other templates. These methods have been used largely for archiving designs toward providing a basis for redesign or collaboration, the design context remaining constant. CDIS architecture can provide a more interdisciplinary aspect to these design storage methods through content indexing augmented by the hierarchical thesaurus.

An interactive mode of operation has replaced the traditional case-based paradigm of question and answer, but some desirable functionality of case-based reasoning has been lost: directly applying design experience to new designs. Extensions to the system to let designers navigate through the system, collect information from relevant design cases, and store pointers to this information in compound documents should be explored. By establishing a mathematical basis for much of this information, new concept evaluation can be done by collecting artifact design models and functional objectives from interesting cases and applying them to a symbol level design evaluation tool (e.g. DesignSheet, developed by Rockwell International (Gonda et al., 1992, Buckley et al., 1992).

Linking active design representations into other network services is also an important extension to the system, providing a means of direct iteration on previous design cases. Bradley and Agogino (1991a) have demonstrated a normative design methodology which can operate on mathematical models to determine the best course of information collecting activity based on possible design alternatives - Intelligent Real-Time Design. This is an important tool for directing information gathering efforts of the user both within the design case base and throughout other internet resources. An example of such a tool applied to catalog component selection (including evaluation under design model uncertainty or imprecision) is described in (Bradley and Agogino, 1991b, Bradley 1993).

A final extension of particular importance to concurrent design again takes cues from the world of architectural. Kunz and Rittal (1970) espoused a view of conceptual design in architecture in which argumentation is the primary mode of resolving the multiple, competing design objectives of multiple stakeholders. Concurrent engineering involves similar concerns among a multifunctional team of engineers. CDIS architecture can be extended to become a multimedia bulletin board which stores a map of arguments surrounding issues in a design. These arguments, including multimedia annotations, would be indexed through the content based WAIS index that is part of the current architecture of CDIS. Thus, case studies will become active, evolving documents instead of retrospective and will promote communication among design team members.

CDIS provides an interface to the design information vital in modern concurrent design methods. The focus on case based support for the conceptual design phase effectively leverages the experience gained among diverse industries and adopted as 'best practices' throughout. The effectiveness of the tool is based on a richly communicative storage medium - hypermedia - and the ability to index the information stored in it at levels of abstraction that make sense throughout the process of conceptual design. CDIS is a step toward a new generation of networked CAD systems integrating hypermedia design databases.

7. Acknowledgements

We would like to thank our collaborators in the development of multimedia case studies, Stanford and Tuskegee Universities for their valuable efforts through the NSF *Synthesis* Engineering Education Coalition. This work has also been supported in part by NSF Grant #DDM-9300025, The Conceptual Design Database. In addition we would like to thank our industrial partners SUN Microsystems, Autodesk Inc., John Wiley, WAIS Inc., and Rockwell International not only for financial and equipment support but for valuable collaboration.

8. References

Agogino, A.M., Hsi, S., 1993a, "Use of Multimedia technology in Teaching Engineering Design", *Proceedings of HCI International '93*, 5th International Conference on Human-Computer Interaction jointly with 9th Symposium on Human Interface, Orlando Florida, pp. 778-783.

Agogino, A.M., Hsi, S., 1993b, "Navigational Issues in Multimedia Case Studies of Engineering Design", *Proceedings of HCI International '93*, 5th International Conference on Human-Computer Interaction jointly with 9th Symposium on Human Interface, Orlando Florida, pp. 764-769.

Agogino, A.M., Sheppard, S., Harris, J., Genalo, L., Mink, K., Krishnagopalan, J., Martin, D., and Saylor, J., 1993, "National Engineering Education Delivery System (NEEDS)", *Proceedings of the Frontiers in Education '93 Conference*, Grayson, L.P., ed., IEEE and ASEE.

Agogino, A.M., Sheppard, S., Oladipupo, B., 1992, "Making Connections During the First Two Years", *Frontiers in Education toward 2000*, Grayson, L.P., ed., IEEE, pp. 563-569.

Bradley, 1993, *Design Optimization under Resource Constraints*, PhD Dissertation, University of California at Berkeley.

Bradley, S., and Agogino, A. M., 1991a, "Intelligent Real Time Design: Application to Prototype Selection", *Artificial Intelligence in Design '91*, (ed. J. Gero), Butterworth-Heinemann Publishers, Oxford, pp. 815-837.

Bradley, S. and Agogino, A. M., 1991b, "An Intelligent Real Time Design Methodology for Catalog Selection," *ASME'91 Design Theory and Methods*, ASME DE-Vol. 31, pp. 201-208.

Bradley, S. and Agogino, A. M., 1991c, "Design Capture and Information Management for Concurrent Engineering", *The International Journal of Systems Automation: Research and Applications,* Vol. 1, No. 2, pp. 117-141.

Bradley, S., Agogino, A. M., and Wood, W.H.,1994, "Intelligent Engineering Component

Catalogs", to appear in *Artificial Intelligence in Design '94*, (ed. J. Gero), Cambridge University Press, Cambridge

Buckley, M., K. Fertig and D. Smith, 1992, "Design Sheet: An Environment Facilitating Flexible Trade Studies During Conceptual Design," AIAA 92-1191 (1992 Aerospace Design Conference), American Institute for Aeronautics and Astronautics, Washington D.C.

Carlstrom, C. M., 1993, "Development, Testing and Assessment of the Cyclone Grinder Multimedia Case Study," MS Project Report, Department of Mechanical Engineering, University of California at Berkeley 94720.

Dixon, J.R., Orelup, M.F., Welch, R.V., 1992, "Computer-Based Models and Representations for Mechanical Design: A Research Progress Report", *Proceedings of the 1992 NSF Design and Manufacturing Systems Conference*, Atlanta Georgia, January 8 - 10, Society of Manufacturing Engineers, Dearborn Michigan.

Dixon, J.R., Orelup, M.F., Welch, R.V., 1993, "A Research Progress Report: Robust Parametric Designs and Conceptual Design Models", *Proceedings of the 1993 NSF Design and Manufacturing Systems Conference*,Charlotte, North Carolina, January 6-8, Society of Manufacturing Engineers, Dearborn Michigan.

Domeshek, E. and Kolodner, J., 1993, "Using the Points of Large Cases", *Artificial Intelligence for Engineering Design, Analysis, and Manufacture*, Vol. 7, No. 2, pp. 87-96.

Garcia, A.C.B., and Howard, H.C., 1992, "Acquiring Design Knowledge Through Design Decision Justification", *Artificial Intelligence for Engineering Design, Analysis, and Manufacture*, Vol. 6, No. 1, pp. 59-71.

Gonda, M., K. Fertig and R. Teeter, 1992, "Aerospace Conceptual Vehicle Design Using an Intelligent Design and Analysis Environment: Design Sheet", AIAA Aircraft Design Systems Meeting, Hilton Head, SC.

Hauge, P.L, and Stauffer, L.A., 1993, "ELK: A Method for Eliciting Knowledge from Customers", *Proceedings of Design Theory and Methodology - DTM '93*, DE-Vol. 53, ASME, pp. 73-81.

Hauser, J., and D. Clausing, 1988, "The House of Quality," *Harvard Business Review*, vol. 66, no. 3, pp. 63-73.

Hennesy, D., and Hinkle, D., 1992, "Applying Case-Based Reasoning to Autocalve Loading", *IEEE Expert*, October 1992, pp. 21-26.

Howard, H.C., Wang, J., Daube, F., and Rafiq, T., 1989, "Applying Design-Dependent Knowledge in Structural Design", *Artificial Intelligence in Engineering Design and Manufacturing*, Vol. 3, No. 2, pp. 111-123.

Howe, A.E., Cohen, P.R., Dixon, J.R., and Simmons, M.K., 1986, "Dominic: A Domain-Independent Program for Mechanical Engineering Design", *The International Journal for Artificial Intelligence in Engineering*, Vol. 1 No. 1, pp. 23-28.

Hua, K., and Faltings, B., 1993, "Exploring Case-Based Building design - CADRE", *Artificial Intelligence for Engineering Design, Analysis, and Manufacture*, Vol. 7, No. 2, pp. 135-143.

Institute for Defense Analysis (IDA), 1988, "Unified Life Cycle Design," Technical Reports.

Jamalabad, V.R., Langrana, N.A., 1993, "Extraction and Reuse of Design Information", *Proceedings of Design Theory and Methodology - DTM '93*, DE-Vol. 53, ASME, pp. 129-138

Kahle, B., Morris, H., Goldman, J., Erickson, T., Curran, J., 1992, "Interfaces for Distributed Systems of Information Servers", Available via anonymous ftp: /pub/wais/wais-discussion/Interfaces.txt@quake.think.com or WAIS server wais-discussion-archives.src.

Kunz, W. and H. Rittel, "Issues as Elements of Information Systems," Studiengruppe Fuer Systemforschung, Heidelberg, Working Paper #131, 1970. (Also described in D. Grant, "How to Use the IBIS as a Procedure for Deliberation and Argument in Environmental Design Planning," *Design Methods and Theories*, vol. 11, no. 4, pp. 185-220, 1977.)

Lehto, M.R., Zhu, W., 1992, "Provision of Reference Materials to Designers via Hypertext", *Proceedings of the 1992 NSF Design and Manufacturing Systems Conference*, Atlanta Georgia, January 8 - 10, Society of Manufacturing Engineers, Dearborn Michigan.

Maher, M.L., and Zhang, D.M., 1993, "CADSYN: A Case-Based Design Process Model", *Artificial Intelligence for Engineering Design, Analysis, and Manufacture*, Vol. 7, No. 2, pp. 97-110.

Maher, M.L., and Zhang, D.M., 1991, “CADSYN: Using Case and Decomposition Knowledge for Design Synthesis”, *AI in Design ‘91*, Gero, J.S., ed. Butterworth-Heinemann: Oxford, pp. 137-150

Morgan, B., Jacobsen, S., 1989, “Image Database Project Summary”, Report, University of California at Berkeley, Information Systems and Technology, Advanced Technology Planning.

National Research Council, 1991,*Improving Engineering Design: Designing for Competitive Advantage*, National Academy Press.

Navinchandra, D., 1988, “Behavioral Synthesis in CADET, a Case-Based Design Tool”, *Proceedings of the DARPA Workshop on Case-based Reasoning*, Kolodner, J., Ed., Morgan-Kaufman, pp. 286-301.

O’Conner, L.J., Lan, M.S., Partridge, D.R., Lee, J.M.F., 1992, “A Case-Based Approach to Automated Weld-Process Design”, *Applied Artificial Intelligence*, Vol. 6, pp. 315-330.

Pearce, M., Goel, A.K., Kolodner, J.L., Zimring, C., Sentosa, L., and Billington, R., 1992, “Case-Based Design Support, A Case Study in Architectural Design”, *IEEE Expert*, October 1992, pp. 14-20

Petroski, H., 1993, “Design Paradigms”, *Proceedings of the 1993 NSF Design and Manufacturing Systems Conference*, Charlotte, North Carolina, January 6-8, Society of Manufacturing Engineers, Dearborn Michigan.

Salzberg, S., and M. Watkins,1990, “Managing Information for Concurrent Engineering: Challenges and Barriers,” *Research on Engineering Design*, vol. 2, no. 1.

Stauffer, L, D. Ullman and T. Dietterich, 1987, “Protocol Analysis of Mechanical Engineering Design,” in *Proceedings of 1987 International Conference on Engineering Design (ICED)*, August, pp. 68-73.

Stauffer, L. and D. G. Ullman, 1988, “A Comparison of the Results of Empirical Studies into the Mechanical Design Process,” *Design Studies*, vol. 9, no. 2, pp. 107-114.

Toye, G., Cutkosky, M., Leifer, L.J., Tenenbaum, J.M., and Glicksman, J., 1993, “SHARE: A Methodology and Environment for Collaborative Product Development”, to appear in *Post-Proceedings of the Second Workshop on Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE.

Ullman, D.G., 1993, “The Evolution of Function and Behavior During Mechanical Design”, *Proceedings of Design Theory and Methodology - DTM ‘93*, DE-Vol. 53, ASME.

Zhao, F. and M.L. Maher, 1988, “Using Analogical Reasoning to Design Buildings”, *Engineering with Computers*, Vol. 4, pp. 107 - 122.

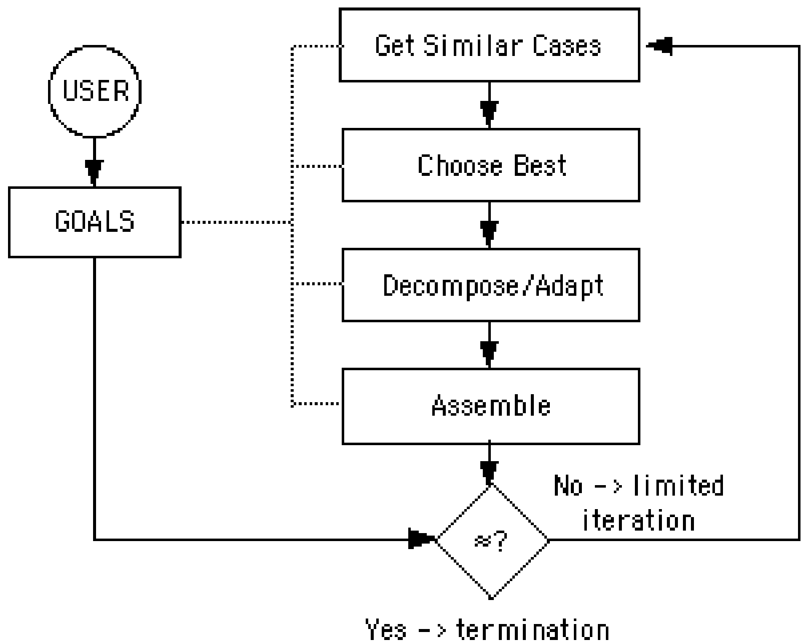

Figure 1 Typical flow of information and control in a case-based reasoning system. The user specifies goals (or perhaps a partial solution - the two are not necessarily stored separately), cases are retrieved, the closest candidates selected, these candidates are decomposed to match new goals, solutions are reassembled, and more cases can be consulted to finish the solution.

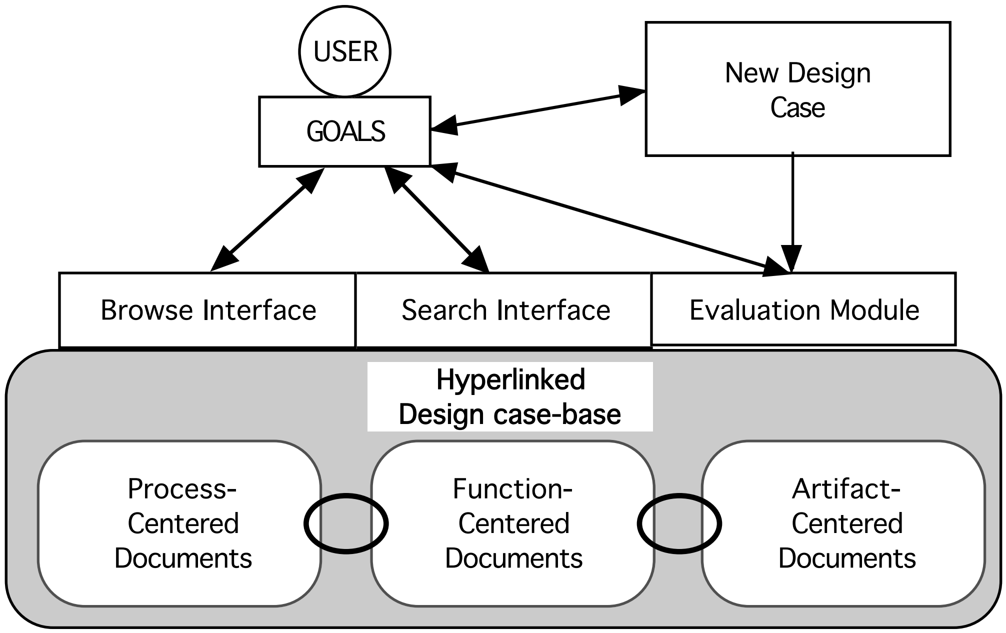

Figure 2 New case-based paradigm: the user is now interacting with browsing, searching, and evaluating subsystems. Partial solutions, rationale, models and objectives are collected through hyperlink storage in a new design document and evaluated.

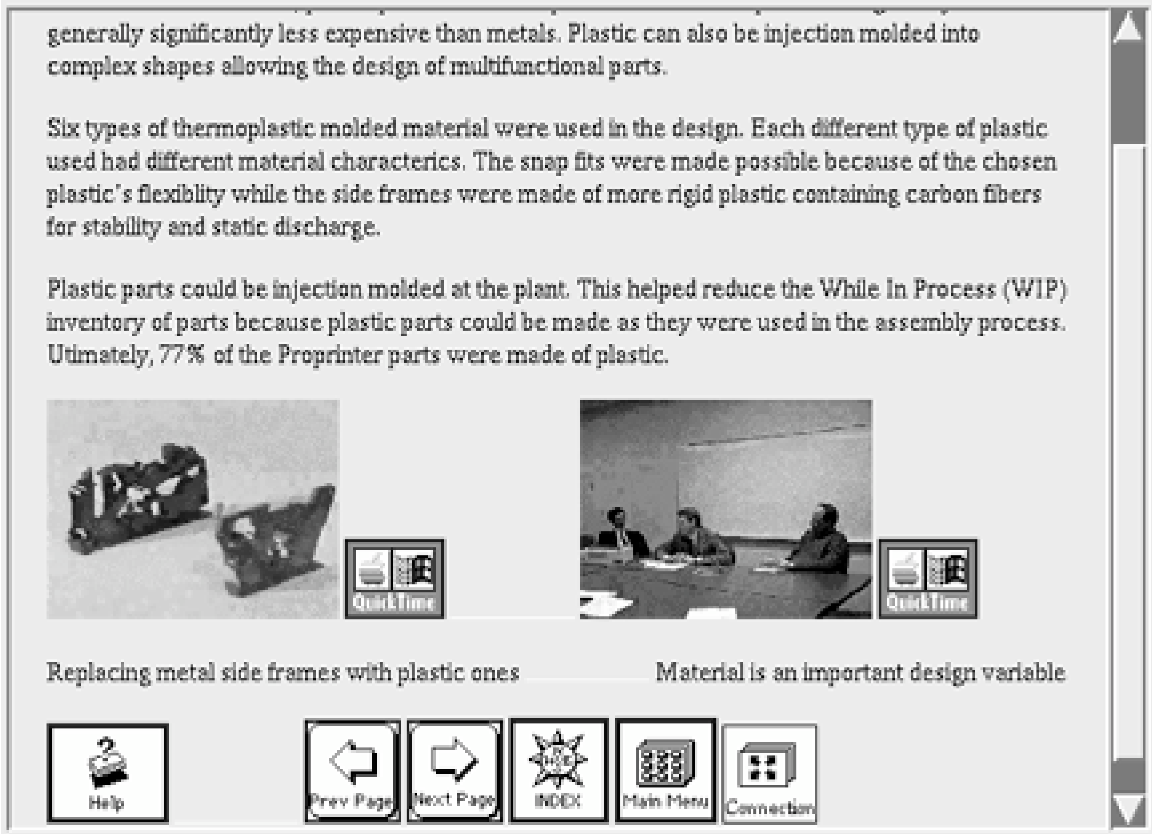

Figure 3 Mosaic interface to an element of the IBM Proprinter case base. The format of this element is typical of the presentations within the case base, synthesizing several information sources into a pointed presentation of a specific topic. Additional hyperlinks are included for navigating around 'local' documents, within the case study through an index or main menu, or outside of the case study through the 'connections' link.

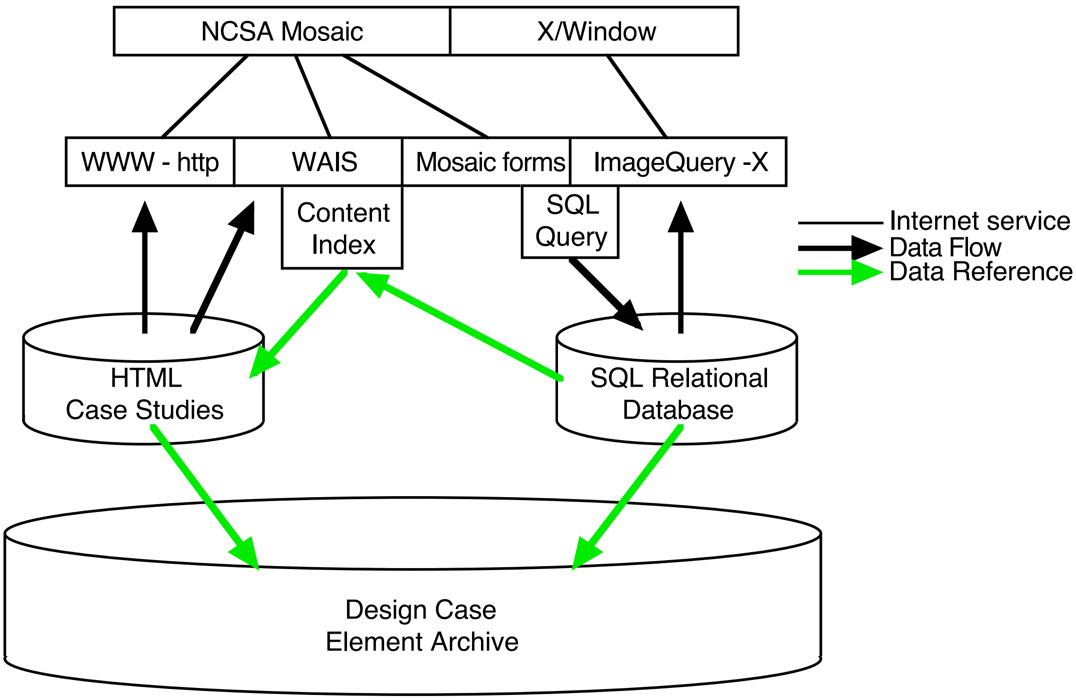

Figure 4 The conceptual design information server includes client service to ImageQuery, WWW, and WAIS. Included in the HTML compound documents served by http are interfaces to WAIS indexes and the SQL database.

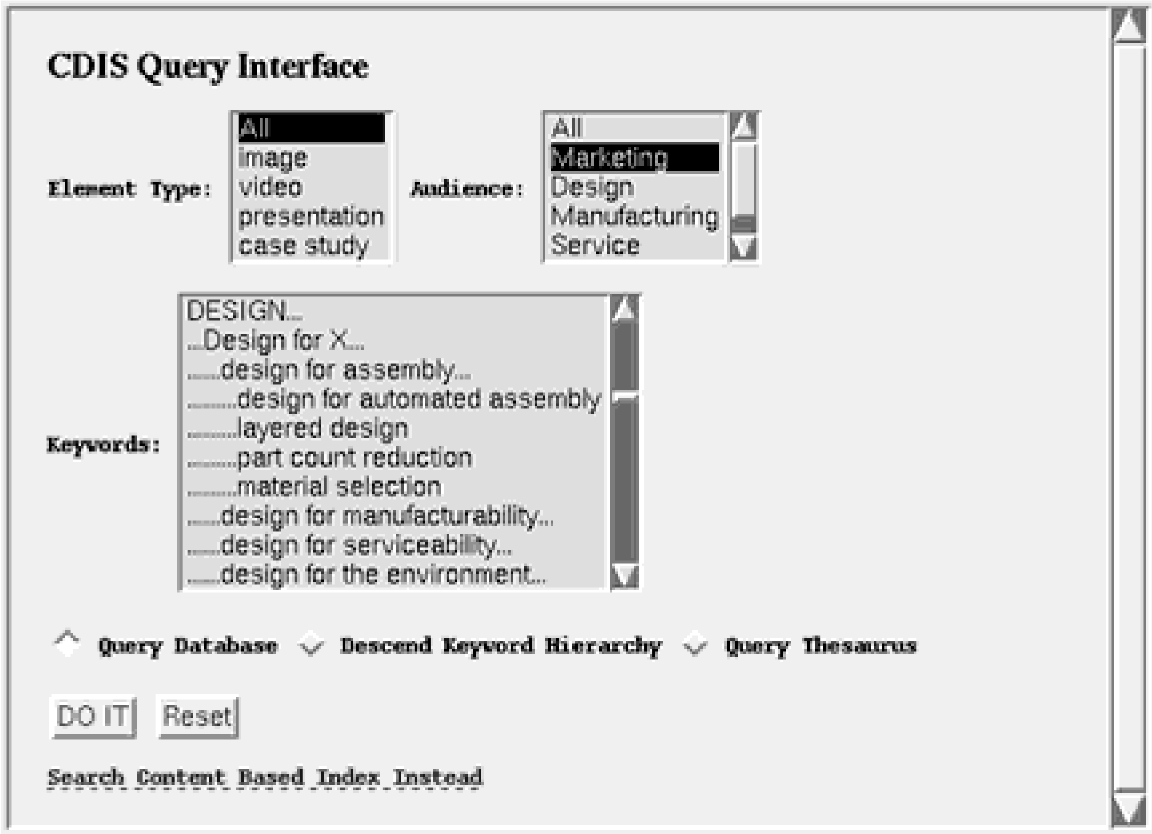

Figure 5 Mosaic interface to the CDIS SQL database. The user is provided means of restricting the information content type to specific formats or audiences. The primary feature is the keyword search interface wherein the user can search the hierarchical ontology for appropriate keywords, referencing the thesaurus for clarification, and query for items indexed on the selected keyword(s). A link to the WAIS free-text index is also provided.

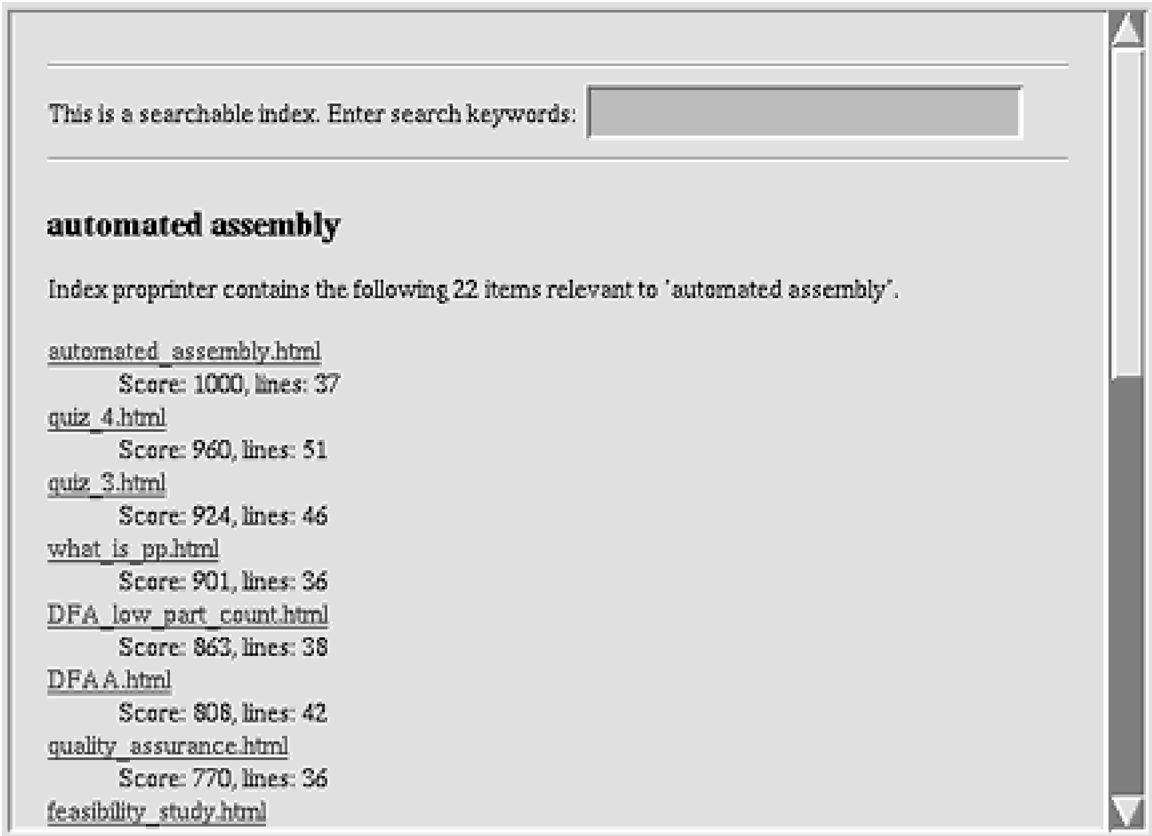

Figure 6 Typical result set from the CDIS WAIS index. The server responds to free-text information requests with a scored list of WWW links to case base elements. From this screen the user can begin browsing the case 'web' simply by selecting a promising-looking document.