

DETC97/CIE-4292

ACTIVE CATALOG: SEARCHING AND USING CATALOG INFORMATION IN INTERNET-BASED DESIGN

S. Ringo Ling

Information Sciences Institute
University of Southern California
4676 Admiralty Way Marina del Rey, California 90292
Email: ling@isi.edu

Jihie Kim

Information Sciences Institute
University of Southern California
4676 Admiralty Way Marina del Rey, California 90292
Email: jihie@isi.edu

Peter Will

Information Sciences Institute
University of Southern California
4676 Admiralty Way Marina del Rey, California 90292
Email: will@isi.edu

Ping Luo

Inquiry.com Inc.,
1730 South El Camino Real, 5th
San Mateo, California 94402
Email: ping@apollo.inquiry.com

ABSTRACT

The exponential growth of the Internet and the increasing communication and computational power have created many opportunities for advancing engineering, manufacturing and business activities. Active Catalog brings a conceptually new idea to design and electronic commerce by providing new forms of catalog information about engineering parts and physical objects. Active Catalog is designed for finding and using information in building design applications in the context of heterogeneous, Internet-based distributed computing environments. The information provided by Active Catalog is active, dynamic, composable, multi-dimensional, share-able, semantically accessible, interoperable and executable in distributed environments.

Active Catalog allows engineers and designers to search for information about catalog parts syntactically and semantically, use that information to construct simulation programs rapidly, and test out their designs on the Internet before committing to actual fabrication of physical systems. It supports a new "Try Before You Buy" paradigm in the Internet-based design and electronic commerce.

This paper presents a number of research issues and our approach in building an Internet-based catalog information resource. The paper describes an initial prototype toward realizing the "Try Before You Buy" paradigm, in the context of the electromechanical design of pump and motor systems. The prototype contains a pump ontology and a search engine for finding seman-

tically equivalent parts. The paper also describes a distributed simulation, which runs on various machines, and exchanges dynamic information using commercial data exchange mechanisms.

INTRODUCTION

Today's designer is faced with increasing pressure on several fronts simultaneously: time-to-market, product cost, product reliability, design for global markets, design for manufacturability, design for re-cyclability and design for the other "ilities". Critical to satisfaction of these goals or constraints is the capability of using proven components and/or sub-assemblies in the system design. The notion of design for the "ilities" via component re-use applies to all of the disciplines found in modern product design. The software designer is following the lead of the mechanical and electrical engineers in following this philosophy with the notions of software backplanes and software workbench ideas. The designer using a re-use methodology needs access to libraries of components with properly and fully documented properties. In addition, the modern designer needs a set of modeling tools in order to prove, in software simulation, that the design is a good one (i.e. one meeting the "ilities"). This reduces the time and cost of doing the design and usu-

ally results also in less costly design. The mechanical designer, in particular, needs simulation systems in addition to the more usual drafting system. Design environments must provide the designer with, in addition to traditional tools, libraries and catalogs of reusable components in relevant design domains (Will 1991). AUTODESK's Parts Library (PartSpec) for drawings of mechanical parts and Smart-Model Library (PartSpec) for circuit design are examples. Currently the Internet offers the possibility of the designer replacing his/her set of filing cabinets of component literature with a rich web of component data provided electronically. The current Internet offerings are mostly electronic versions of what is available on paper (still very valuable to the designer) but more functionality is possible and is expected.

Most of the benefits of catalogs and libraries are based on a major prerequisite: that the information is easily and conveniently accessible to the designer (i.e., it is reachable and usable). This assumption does not naturally hold in general. Although rich in content, today's libraries are poor in access and their contents are designed to be consumable by a human reading a document or looking at an image. The design domain is one where the information is to be automatically consumed by a computer and is harder. This poor situation would be improved a) by using domain knowledge to give designers easier access to library contents during the course of design and b) by improving the means of consumption of the information by computers after it is retrieved.

There exists a gap between the high-level, conceptual mental model of a part needed by a designer and the low-level, physical query that retrieves the needed information from a library. Too often, the designer is required mentally to map the description of a desirable component for solving a particular problem onto a set of searchable attributes recognizable by the library/catalog. That is, the user needs essentially to understand the "schema" of the libraries and the searchable technical attributes of the library contents before mapping the need into a set of queries. This mapping is not trivial and imposes a major mental burden on designers, deflecting them from focusing directly on solving their design problem. Using the information from the library/catalog is equally onerous and often involves manually transcribing the information into the local design tool environment. This is especially true when the data must be entered into CAD or be prepared for consumption by distributed simulation environments.

In this paper, we describe a prototype system, Active Catalog, that addresses the above problems by using a rich body of domain knowledge to describe the library components, adding to and annotating traditional library information and aiding the use of the retrieved information in

CAD applications. Designers, using the system, can search library contents using the familiar terminology of the design domain. A search engine, based on the knowledge of the domain and the meaning of the queries specified in the domain vocabulary, accesses library components that satisfy the functionality meant by the queries, rather than by matching the keywords or synonyms in the queries. Active Catalog, suggested by its name, adds to traditional library contents (such as textual and image information), a rich set of models of different types that, in turn, allow a designer to evaluate dynamic and behavioral aspects of a component in a system via interactive simulation built from or using down-loaded models, viewers or even applets of simulation code.

The goal of this research is to improve both design environments and the general consumption of information found in catalogs and libraries. We approach this goal, done here in the context of electro-mechanical engineering design, by addressing several fundamental human computer interface, cognitive issues. As noted, there can be a gap between the user's mental model of a part and the set of queries needed to access it. We humans cannot perform high quality reasoning and provide correct judgment if there exists no direct mapping between a real world object and our mental model of that object; and we are much less productive when we are diverted by too many distracting tasks; we cannot keep track of many threads or recall all what we know. Last, but not least, we cannot access material if we are not aware of its existence. We aim at providing technologies that enable designers to dedicate their mental resources to the design problems at hand and are using a knowledge-based approach to do it.

The rest of the paper is organized as follows. We first define Active Catalog by a list of its design objectives. We then present the key relevant system components. These components include an information search environment that enables a high-level approach to information query, and a knowledge-based search mediator that hides distributed and heterogeneous information sources from a user. Also, we describe an information consumption environment to support distributed simulations in evaluating the dynamic and behavioral aspects of a piece of information retrieved. Finally, we describe the related work and conclude the paper.

1 ISSUES AND APPROACH IN ACTIVE CATALOG

Active Catalog is designed for both *finding* and *using* information in building design applications in the context of heterogeneous, Internet-based distributed computing environments. Its design requirements are driven by its vision and its computing environments in which the vision will be realized, i.e., finding and using information in building

distributed applications and engineering simulations.

To fulfill the vision, Active Catalog must provide *active* and *dynamic* information. Information must not only be responsive to user queries, but also can be easily used by computers in simulations. Information has to be *composable* so that it can be *integrated* and reused in building large scale systems and applications. Information must represent *multi-dimensional* aspects of physical parts and objects so that it can be used and shared in multiple domain environments. Information can be *searched* and accessed in a number of ways, using keywords or functional descriptions. Information must be *interoperable* and *executable* in heterogeneous distributed environments.

Together, the above keywords defines a set of design objectives of Active Catalog. We will examine these objectives in the following sections.

Information is Active and Dynamic

Information in Active Catalog is used not only for human understanding, but also for computer consumption. In particular, information can be directly used by computers as modules in simulation programs. A consequence of that requirement is that information is not only static data but also dynamic program modules. With the exception of Java Applets, the current Web-based information is data in various text, image, audio or video formats, such as HTML or GIF formats etc. The data is annotated with format commands which tells a Web browser how to display the data on a computer screen for human visual and audio consumption.

The current Web information has served the human information needs very well, and fueled the recent explosive growth of Web-based activities. However, the information is essentially static data, primarily used for direct presentation of information content to users. The data, by itself, cannot be executed at user machines. It has to be processed and interpreted by some computer programs. If users want to use the information in a HTML file that they find on the Web, they need to extract the data from the HTML file and re-enter it into the programs that they need.

Active Catalog provides not only data, but also program modules. The information is active, because the program modules can be directly executable on user machines without any intervention from users. The information is dynamic, as the program modules can be executed in user environments to meet users special needs. The information can also respond to current situations and changing circumstances. A Java applet is an example of this active and dynamic information. However, Active Catalog's information is more than Java applets, as described in the next section.

Information is Composable and can be Integrated

Java applets can be executed on user machines, but they are essentially stand-alone applications. For example, Java applets are primarily used for animations in Web documents. They have only limited interactions with user environments, such as using user processors or accepting user interactive inputs. The restrictions are due to security concerns and the underlying assumption that applets cannot be trusted in general. While the concerns may be valid in some cases, the security concerns prevent applets from being widely used in user programs.

Active Catalog is based on the assumption that information will be shared and used in a co-operative environment. Its objective is to support designers and engineers in their work. Information can be down-loaded from a few but trust-able sites, such as the sites of software vendors or colleagues and designers can incorporate the down-loaded information, i.e., programming modules, directly into their programs. Instead of writing programming codes, designers can download well-tested and robust modules that suit their purposes and assemble them together. Building a large scale programming system is feasible. Instead of the process of writing and debugging program codes, building an application becomes the process of selecting and assembling program modules.

For modules to be easily composable and integrated in applications, they must be modular and have well-defined interfaces. Modularity allows easily mix of modules, while well-defined interfaces reinforce consistent composition of programming modules. The issues of modularity and interfaces have received a lot of attention in the areas of software reuse and object-oriented programming language. However, their focus is on software components and syntactic composition. The support of semantics is limited other than using the documentation to express the intents of programming modules. Active Catalog build on the existing strength of software reuse and object-oriented programming. In addition, Active Catalog focuses on ontologies and representations of physical objects to support semantic composition and integration.

Information is Multi-Dimensional and Share-able in Multiple Domains.

Active Catalog information is not just software modules. It represents features and behavior of physical and engineering parts. Because physical parts interact in multiple domains, Active Catalog information is multi-dimensional. It will be shared and used in multiple domains.

A physical object has features and behavior in multiple domains. For example, a pump is a fluid mechanical device to deliver fluid at certain flow rates and pressure heads.

However, a pump also accepts mechanical power, such as torque and RPM (revolution per minutes) from its mechanical shaft or coupling. A pump has structural information, such as its pipe threads and diameters. The pipe diameters and threads constraint a pump connection to other pipes. Finally, a pump also has its geometric dimensions and weight, which can affect the physical location of the pump.

Not only does a physical object exhibit behavior in multiple domains, it can interact with other objects in multiple domains too. For example, the fluid mechanical behavior of a centrifugal pump is driven by mechanical energy from its pump shaft. The mechanical energy may be provided by an electrical motor, which converts electrical energy into mechanical energy for the pump. In addition, a pump may generate vibration due to its rotation. If the pump is not properly supported by appropriate frames, it can create vibration and noise. One behavior in one domain can lead to behavior in other domains. In engineering design, engineers have to consider all these behavior and their interaction all the time. One of the challenges in Active Catalog is to represent all these multi-disciplinary behavior in an integrated fashion and allow the behavior to be shared in multiple domains.

Active Catalog provides domain ontologies for classifying physical objects. The interaction between objects from different domains will be guided by a shared ontology and mapping functions across domain ontologies.

Information is Accessible Syntactically and Semantically

Before we can use Active Catalog information, we have to find it. Finding information is complicated by the fact that the information is distributed throughout the Internet. Currently, there is a lot of research focusing on finding information in the Internet. Many Internet-based services, such as Yahoo and Lycos, exist for finding Web and HTML documents. Also many on-line catalogs services provide interfaces to their databases through CGI and users can perform relational database queries. While these search mechanisms are useful, they primarily rely on keyword search, coupled with statistical correlation techniques to identify potential matches. Active Catalog makes use of these mechanisms. However, Active Catalog also exploits certain features of physical objects to provide additional search capabilities. The features are: physical objects are structured information, and they are designed for certain functional purposes.

Information about physical objects is structured. While a physical object may have many attributes, the attributes are structured information. An attribute is defined in a domain and has certain relationships with other attributes. Often there are precise definitions and laws governing the

values and relationships of these attributes. Objects and their attributes can be organized in certain information structures, such as hierarchical trees or semantic networks. Active Catalog can exploit the information structures to organize its search.

Physical objects are designed for certain purposes and have functional specifications. The main function of a pump is to deliver fluid at certain pressure head and flow rate. This functional description can provide a powerful way to search for equivalent objects serving similar functions, without the use of keywords or equipment names. For example, a query description has the statement of "delivering fluid at a certain pressure", and Active Catalog can propose pumps and compressors, even though pump or compressor is not mentioned in the user query.

Active Catalog uses domain ontologies to organize physical objects to facilitate its search. In addition, objects are provided with functional descriptions such that semantically equivalent objects will be identified.

Information is Interoperable and Executable in Distributed Environments.

Active Catalog information is used in heterogeneous and network-based distributed environments. Information can be located at different information sources. After a program is developed, it may be executed on a different machine over the network. The key issue is: how to execute computation in distributed fashion.

Interoperation among software modules in heterogeneous environments has been a major issue in distributed systems. Unix Sockets, RPC (Remote Procedural calls), DDE (Dynamic Data Exchange) are earlier work to deal with that issue. OLE, CORBA and Java are current technologies developed for the Internet-based distributed environments. Active Catalog uses a number of approach to handle heterogeneous issues: by representing information in a platform independent fashion, and by using interface and wrapping agents for interoperating software modules. For example, Active Catalog may represent software modules in Java, a platform independent programming language. Java is developed by Sun for program developments in heterogeneous and network-based environments. A Java program is translated into bytecodes, an architecture neutral intermediate format. The bytecodes can be executed on any system as long as a Java virtual machine exists for that system. The differences in machines and operating systems are shielded from users by the architecture neutral Java language and the Java virtual machine. In the cases where modules exist in different languages, Active Catalog may interoperate the modules by wrapping the modules and using CORBA, a common interface developed for distributed environment

to inter-operate between them.

After a program module is built, it may be executed on machines across network. Because modules are in Java, they can be executed on different machines. Furthermore, the multi-thread nature of Java can facilitate the parallel computation and synchronization of Active Catalog modules. Active Catalog will also make use of current research in high performance computation, such as PVM (Parallel Virtual Machine) (Geist et.al. 1994) MPI (Message Passing Interface) (Gropp et.al. 1994), and load balancing mechanisms to distribute parallel executions. Furthermore, an intelligent scheduling mechanism based on contract nets for negotiating and allocating computing resource is planned.

In the next two sections, we will describe how to find catalog information and how to use it in engineering design. The current architecture of Active Catalog is shown in Figure 1. The discussion will be in the context of electro-mechanical engineering design. Pumps and motors are used as examples throughout the remainder of this paper. The pump domain was chosen in the context of simulation based design of electro-mechanical systems in ship design and maintenance. Although the description of the work is specific to the pump and motor systems, the technologies and mechanisms are domain independent and have wider applicability. In fact, the Active Catalog is part of an integrated system where engineering design, purchasing, logistics and electronic commerce will be tied together. We will describe its current implementation and future work.

SEARCHING FOR CATALOG INFORMATION IN DESIGN

Searching for Internet-based catalog information is provided in a user-centered information environment where a user can issue high-level conceptual queries or partially specified queries for the desirable information without knowing the details of how to access the objects, i.e., what the technical terms for the objects are, and where they are located in the system. A knowledge-based search engine, taking the user's high-level query as input and the domain knowledge base, matches catalog contents with the semantics of the query. Beside searching for information using queries, the user can also browse catalog contents or use multi-level pull-down menus to access the catalog objects.

When an object is located, Active Catalog extracts the multi-dimensional information of the object and dispatches it to the corresponding software in the user's environment such as simulators, spread-sheets, helper applications and viewers that handle the appropriate modality. When necessary, a distributed simulation environment can be automatically configured and executed, as described in Section 4.

Pump Ontology and Semantic Network

To facilitate searching and browsing catalog information, Active Catalog employs a set of *domain ontologies* that contain taxonomies, terminologies, and semantic networks. These ontologies capture the true nature of the domains. An ontology describes taxonomies of the domain objects (parts and components in the catalog or library); their attributes, values and value constraints; taxonomies of the attributes, values and value constraints; and relationships among these objects, attributes, and attribute values and value constraints. Example taxonomies for the pump domain are given in Table 1.

A taxonomy is organized as a graph rather than a tree structure to provide flexibility in accessing the objects. The graph structure of the domain objects (and their attributes and values) gives multiple paths among them, and the user of the taxonomy can reach the same object via different paths. This enables the user to issue partially specified queries in terms of known pump types/features without sacrificing the capability of reaching the target. The taxonomy does not require its user to know and remember the exact types of the pumps needed. For example, a Reciprocating, Single Acting, Power Multiplex Pump is reachable via any of (or combination of) the following pump types: Reciprocating Pump, Single Acting Power Pump, Multiplex Pump, and all the pump types that are the parents (in the type hierarchy) of these three types. Also, the hierarchical organization of the objects allows queries at different levels of generality, where queries about higher level classes retrieve results that are super-sets of results from the queries about lower level classes. Taxonomies for attributes and value constraints give similar benefits.

Functional Description Hierarchy
transfer fluid
 transfer corrosive fluid
 transfer non-corrosive fluid
circulate

Ontologies in Active Catalog can have functional descriptions of the objects as attributes to facilitate semantic access to the information. A functional description contains a statement about the purpose of a device. The statement consists of an action, objects involved in the actions and the object attributes. For example, the functional description of a pump is: Increase the pressure head of a fluid to a certain value at a certain flow rate. "Increase" is an action. The object is fluid, and its attributes involved in the functional description are pressure and flow rate.

Given the functional description of the objects and the relationships among the functions, the system creates a template that can be filled with functional descriptions of the desired objects. This is transformed into a query to retrieve

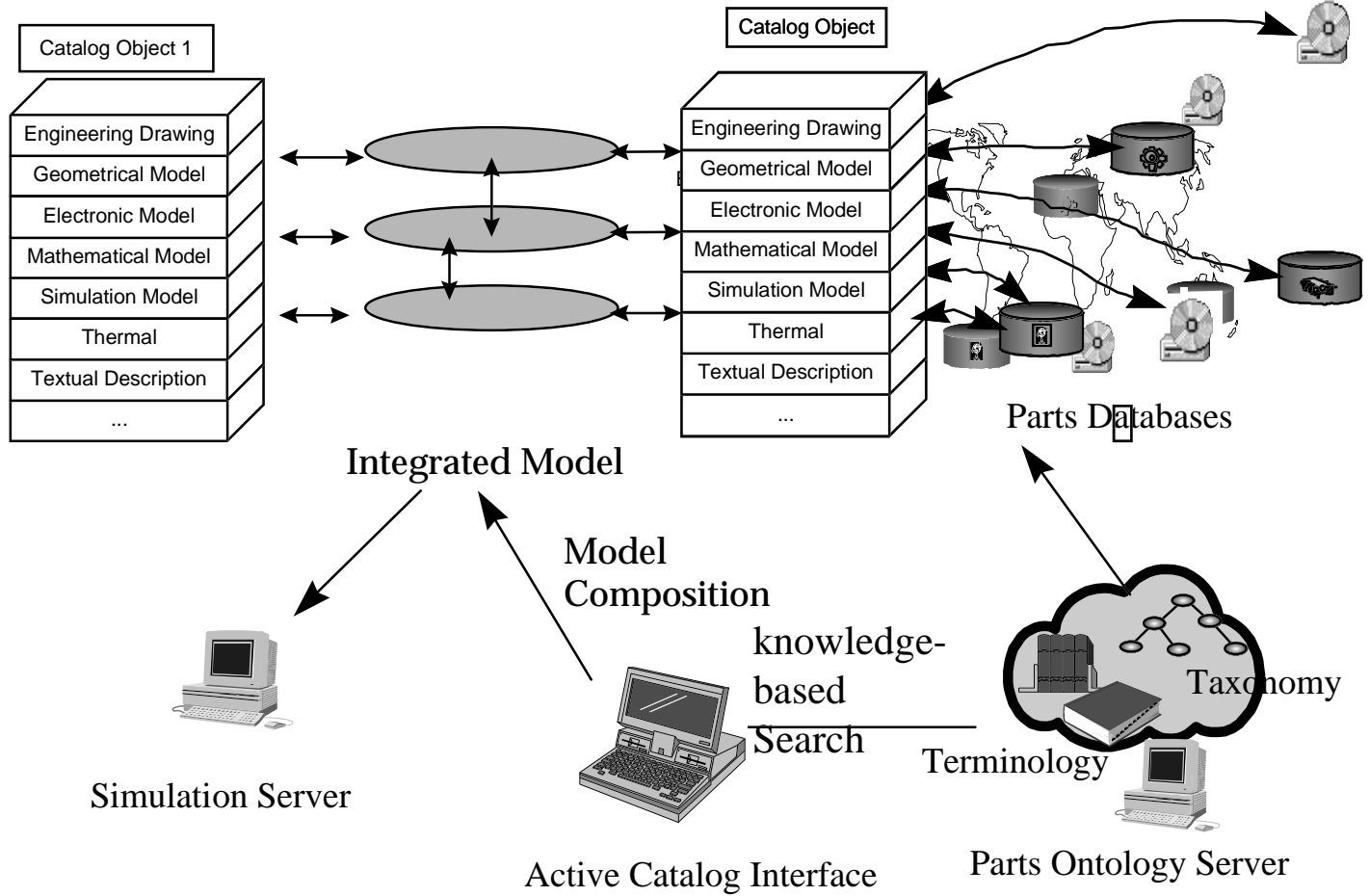


Figure 1. Active Catalog Architecture

the objects that meet the functional descriptions. For example, given the functional structure shown above, a user can access both corrosive and non-corrosive pumps if the user query does not specify the types of fluid. However, if a query involves corrosive fluid, only corrosive pumps will be returned.

Additional knowledge such as semantic network, woven with the taxonomies, declaratively describes a richer set of domain knowledge that is traditionally used by experienced engineers. Figure 2 graphically depicts a piece of the semantic network describing (a part of) functional hierarchy for pump, the relationships between pumps and motors, and the relationships between the type of fluid to be transferred and the casting materials for pump.

This semantic network captures implicit knowledge about applications. For example, the network contains a piece of deductive information that if a pump is used to

transfer corrosive fluid, then the pump's material should be corrosive-resistant. In fact, our network provides different types of corrosive fluid, corrosive resistant materials, and a set of correspond implication rules. Users can then directly describe the use of the pump in application terms, such as finding a high capacity pump for transferring sea water or finding a motor that drives a high-capacity centrifugal pump, without having to know the exact technical terms for describing pump parameters.

Also, the pump ontology and its semantic network has an open architecture. It provides a template for expanding into a much richer and detailed network depending on the need of an application, and the network is sharable across domains.

Currently, the semantic network in Active Catalog is modeled in Loom (MacGregor 1990) (MacGregor 1994), a knowledge representation system language and environ-

Table 1. Sample Taxonomies of Pump Types, Attributes and Value

| Type Hierarchy | Value Constraint Hierarchy | Attribute Hierarchy |
|-------------------------------|---------------------------------|--------------------------------------|
| Centrifugal Pump | Construction | Impeller Structure |
| Axial Flow Centrifugal Pump | Casing Construction | Impeller Structure By Flow Direction |
| Mixed Flow Centrifugal Pump | Coating And Lining Construction | Impeller Structure By Openness |
| Radical Flow Centrifugal Pump | Impeller Construction | |
| Screw Centrifugal Pump | Wearing Ring Construction | |
| Special Pump | Driver | Rotor Structure |
| Electromagnetic Pump | Brake Horsepower | Rotor Structure |
| Gas Lift Pump | Driver Position | Bearing Structure |
| Hydraulic Ram Pump | Driver Type | Circumferential Piston Structure |
| Jet Pump | Pitch | Diaphragm Structure |
| Pitot Pump | Installation | Flexible Member Structure |
| Screw Centrifugal Pump | Geometry | Gear Structure |
| Viscous Drag Pump | Mounting | Lobe Structure |
| | Weight | Piston Structure |
| | | Screw Structure |
| | | Vane Structure |

ment developed at ISI for constructing intelligent applications. The Loom system provides deductive support for the declarative portion of the Loom language. Declarative knowledge in Loom consists of definitions, rules, facts, and default rules. The semantic model of pumps consists mainly of definitions and matches Loom's power. The pump ontology and its semantic network had 300 classes of pumps, 3000 attributes and took three months to build. To support building domain ontologies and incremental development of a shared ontology, Active Catalog is using Ontosaurus (Swartout et. al. 96), a web-based ontology editor.

Knowledge-based Search Engine

Active Catalog provides high level search by insulating users and user queries from the details of the information sources, including the physical locations of the sources, their query language, organization, etc. The semantic model of the above semantic network provides a vocabulary that is close to the one used in an engineer's daily work and a structure that captures the nature of the domain expert knowledge. By providing a mapping between semantic models and information source models, Active Catalog can translate a semantic query into corresponding queries of physical

data repositories. The database that stores the information can be distributed in the Internet. Also, the search engine can access up-to-date information over the net.

Active Catalog uses SIMS (Arens et al. 1993), as its knowledge-based search engine. SIMS is an intermediate layer - a mediator between information sources and human applications programs. Queries to SIMS are in a uniform language, independent of the distribution of information sources, the query languages, and the locations of sources, etc. SIMS determines which data sources to use, how to obtain the desired information, how and where to temporarily store and manipulate it, and how to maintain an acceptable level of performance.

At the top level a query in Active Catalog is translated into a SIMS query using the terminology of the semantic network. SIMS converts the initial query into a set of semantically-equivalent queries to the physical information sources. SIMS needs to know the structure of the information sources and the mappings from semantic terminology into physical data sources. The structural information of the data sources can be modeled easily (in fact, the major part of the model can be generated automatically by querying the schema information of the data sources); currently the KB-DB mapping needs to be hand-crafted.

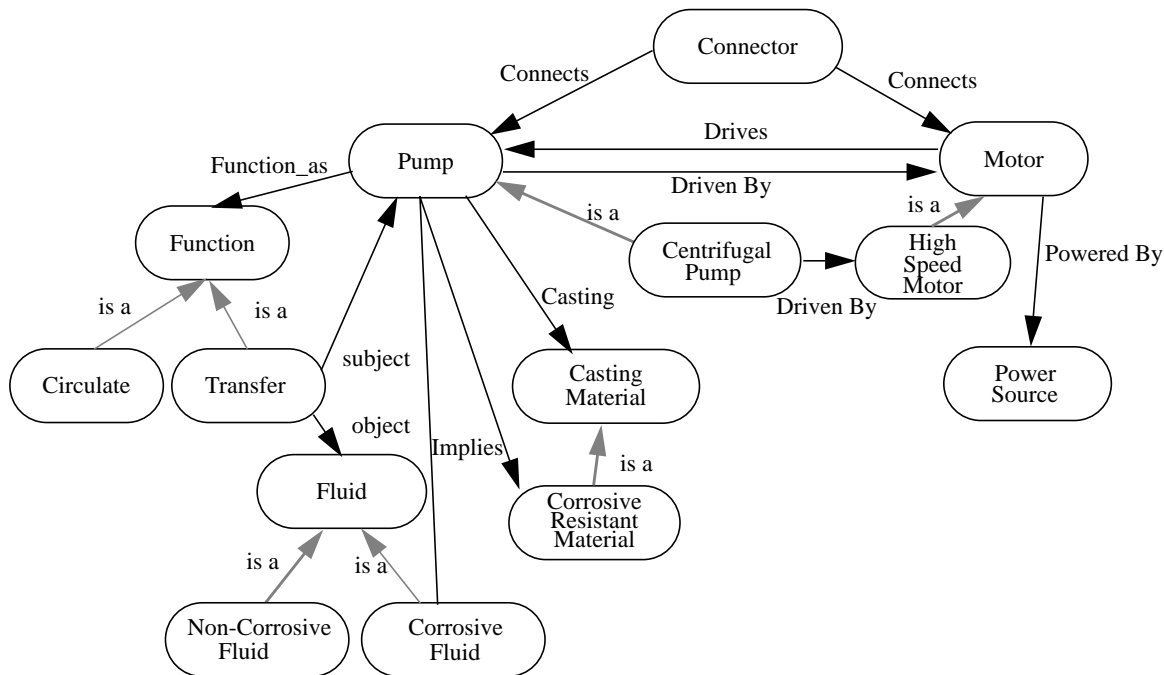


Figure 2. A fraction of Pump Knowledge in Semantic Network

With the three pieces of model, the domain model, the information source schema model, and the KB-DB mapping model, SIMS performs reasoning on a semantic query (such as changing "seawater pump" into "bronze casting material"), partitions a potentially complex semantic query into many small pieces and transforms the small pieces into actual queries to the physical information sources, taking into account the heterogeneous and distributed nature of the sources, the efficiency of database joining, parallel accessing information sources, and redundant information storage such as mirroring and partial overlapping.

USING CATALOG INFORMATION IN DESIGN

Major effort in Active Catalog has been focused on aiding its users (in the example, engineering designers) in evaluating the information returned in response to their semantic queries. When a piece of information is retrieved, based on its type, the system dispatches the information to an appropriate viewer or consumer. For example, if the information is an engineering drawing in AutoCAD (AutoCAD), the system invokes the AutoCAD environment and loads the drawing; if the information is a differential equation or transfer function described by a MatLab model, MatLab/SimuLink is brought up with the model inserted; if

a brochure, the presentation environment is run. The approach is similar to the one used in Netscape Browser where appropriate plug-in viewers are invoked for different MIME data types.

We have a working prototype for the distributed simulation of the pump/motor example. Mathematical and transfer function simulation models of both pump and motor are sent to MatLab/SimuLink for dynamics analysis while kinematic models for simulation of the pump pressure and stroke interactions are sent to WorkingModel, a commercial simulation tool for analyzing Newtonian mechanics of mechanical parts. Each system has its own display and animation features. The working system was implemented on a set of distributed machines running on top of DDE and TCP.

Users can examine the results of the simulation models, select other models, other pumps, or motors, or combinations of both, until an acceptable design solution is reached. The satisfactory parts are then "bought". Currently we are building an agent-based broker middle-ware based on Java and CORBA to interoperate between simulation models. We also plan to integrate with FAST (FAST), an electronic commerce system operating at ISI to implement real purchases.

The Active Catalog system implements "Try Before

You Buy”, a powerful, new paradigm of information access. This paradigm, although illustrated here in the domain of pump/motor systems, is not limited to this particular domain or solely to engineering design. Instead it is applicable to a wide range of applications as diverse as education and electronic commerce.

Related Work

There has been an increasing interest in developing on-line catalogs. IndustryNet (IndustryNet) and PartNet (PartNet) are such examples. These can be federated or centralized databases, depending on the structuring of the provider’s product offering. The search mechanisms of these catalogs rely mostly on keywords, with searchable attributes and values, rather than based on the semantics of queries. Also, the objects are static data. Users have to write their own programs and re-enter the data into their programs.

Ontolingua, the Ontology project at Stanford (Farquhar et al. 1995) provides a framework for building ontology, and unifying and disseminating models, as Ontosaurus does. The system is a knowledge-base independent language designed for ease of translation between different knowledge bases. It is not designed to reason about services, a key in dynamic access to information sources.

A number of other researchers are exploring knowledge-based design environments. In particular, SHADE system (McGuire et. al. 1993) uses an ontology based approach to facilitate collaboration between product development and integrated manufacturing. SHADE is based on Ontolingua while Active Catalog is based on Ontosaurus. The differences between the two ontology servers apply in the same way to the differences between SHADE and Active Catalog. Also, Active Catalog provides more powerful mechanisms for information evaluation and consumption by providing simulations of many modalities, in addition to viewing documents and drawings.

Mediator (Gaines et. al. 1995) is a knowledge support system for managing activities in manufacturing process. Although it provides KL-One like inference services and powerful graphical capabilities, Mediator focuses on accessing remote files and applications rather than on helping users to access, evaluate, and consume information in distributed task environments.

STEP (the Standard for Exchange of Product Model Data) (Fowler 1995) builds an environment for engineering collaboration for engineering data exchange. It can be used for Internet simulations of mechanical systems design. By building Active Catalog on STEP, we can accomplish industrial enterprise integration: concurrent engineering and virtual enterprises.

CONCLUSION

The exponential growth of the Internet and increasing communication and computation power have created many opportunities for advancing engineering, manufacturing and business activities. Active Catalog brings a conceptually new idea to design and electronic commerce activities by providing new forms of catalog information about engineering parts and objects. The information is dynamic, active, composable, multi-dimensional, share-able, semantically accessible, interoperable and executable in the Internet-based, heterogeneous and distributed computing environments.

Active Catalog allows engineers and designers to search for information about catalog parts in flexible ways, use that information to rapidly construct simulation programs, and test out their designs on the Internet before committing to actual fabrication of physical systems. It supports a new “Try Before You Buy” paradigm in the Internet-based design and electronic commerce.

This paper presents our approach and our initial prototype toward realizing that paradigm, in the context of designing electro-mechanical systems of pumps and motors. The paper describes a pump ontology semantic network and a search engine for finding semantically equivalent parts. This paper also describes a distributed simulation, which ran on various machines, and exchanged computational results dynamically using commercial data exchange mechanisms.

The prototype also reveals many research issues and we are pursuing them. They are: building better ontologies for functional search, mappings between domain ontologies for knowledge sharing and multiple domains simulations, explicit representations of simulation modules for model composition, and high level mechanisms for interoperation and information exchange based on CORBA.

ACKNOWLEDGMENT

We would like to thank Ravi Gauba, Houchang Lu and Shivanand Bhajekar for their implementation efforts. This work was supported by DARPA under contract J-FBI-95-159. The contents represent the opinions of the authors, and do not reflect official positions of DARPA or any other government agency.

REFERENCES

Y. Arens, C.Y. Chee, C.N. Hsu and C.A. Knoblock. (1993) *Retrieving and Integrating Data from Multiple Information Sources*, International Journal of Intelligent and Cooperative Information Systems. Vol. 2, No. 2. Pp. 127-158.

R.J. Brachman Ed. (1991) *Principles of Semantic Networks: Explorations in the Representation of Knowledge*, The Morgan Kaufmann Series in Representation and Reasoning.

D. Benjamin, D. Grossman, P. Postel, C. Powley, E. Brodersohn and R. Fadia *Federated Services in Electronic Commerce: Architecture and Issues*, ISI Research Report, ISI/RR-96-448, 1996.

FAST. <http://info.broker.isi.edu/1/fast>.

J. Fowler, *STEP for Data Management, Exchange and Sharing*, 214 pp., Technology Appraisals Ltd., UK, 1995. (ISBN 1-871802-36-9)

B. R. Gaines, D. H. Norrie, and A. Z. Lapsley, *Mediator: an Intelligent Information System Supporting the Virtual Manufacturing Enterprise*, Proceedings of 1995 IEEE International Conference on Systems, Man and Cybernetics. pp.964-969. New York, IEEE.

A. Geist, A. Beguelin, J. Dongarra, *PVM (Parallel Virtual Machine): A User's Guide and Tutorial for Networked Parallel Computing*, MIT Press, December 1994.

W. Gropp, E. Lusk, A. Skjellum, *Using MPI: Portable Parallel Programming with the Message Passing Interface*, MIT Press, October 1994.

T. R. Gruber, *Ontolingua: A mechanism to support portable ontologies*, Stanford University, Knowledge System Laboratory, Technical Report KSL-91-66, March 1992.

IndustryNet. <http://www.industry.net>.

J. G. McGuire, D. R. Kuokka, J. C. Weber, J. M. Tenenbaum, T. T. Gruber, and G. R. Olsen. *SHADE: Technology for Knowledge-Based Collaboration Engineering*, Journal of Concurrent Engineering: Applications and Research (CERA), 1(2), September 1993.

R. M. MacGregor, *A Description Classifier for the Predicate Calculus* In Proceedings of the Twelfth National Conference on Artificial Intelligence, 1994.

PARTNET. <http://www.part.net>.

PartSpec and MaterialSpec. Autodesk, Inc. 111 McInnis Parkway, San Rafael, CA 94903, USA.

W. R. Swartout, R. Patil, K. Knight, and T. Russ. *Toward Distributed Use of Large-Scale Ontologies*, In Proceedings of the Banff Knowledge Acquisition Workshop, Banff, Canada, Nov. 1996.

P.M. Will. *Simulation and Modeling in Early Concept Design: An Industrial Perspective*, Research in Engineering Design. V3, N1. Springer International, 1991

Knowledge Revolution, 66 Bovet Road, Suite 200, San Mateo, CA 94402, USA.