

DETC98/CIE-5521

ACTIVE CATALOGS: INTEGRATED SUPPORT FOR COMPONENT ENGINEERING

Murilo Coutinho¹
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
coutinho@isi.edu

Ragy Eleish
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
ragy@isi.edu

Jihie Kim
USC Information Sciences
Institute
4676 Admiralty Way
Marina del Rey, CA 90292
jihie@isi.edu

Vished Kumar
USC Information Sciences
Institute
4676 Admiralty Way
Marina del Rey, CA 90292
vkumar@isi.edu

S. Ringo Ling
USC Information Sciences
Institute
4676 Admiralty Way
Marina del Rey, CA 90292
ling@isi.edu

Robert Neches
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
rneches@isi.edu

Peter Will
USC Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292
will@isi.edu

ABSTRACT

A recent National Academy of Sciences study shows that design and manufacturing is dominated by three “eighty percent” rules. Eighty percent of the cost of a product typically is the purchase cost of its components. Eighty percent of the manufacturing costs are determined in the first twenty percent of the design phase. And eighty percent of an engineer’s time is spent *not* on design, but on communications: obtaining information about components comprising the design and coordinating with other engineers about design issues. Thus, the speed and quality of product development is heavily dependent on a designer’s ability to perform *component engineering* — to configure a set of components that work well together with respect to performance, reliability, cost and delivery schedule. This paper describes services provided in the Active Catalogs system that support the component engineering process. These services enable a designer not only to find candidate parts to serve as individual components in a design, but to explore and assess interactions between

candidate parts for sets of components. These services include mechanisms for: (1) creating queries for parts based on their intended use rather than merely parametric specifications; (2) refining those queries to take account of constraints imposed by other components; (3) providing multi-modal information to help designers assess and compare candidate parts; and (4) generating simulation models of candidate parts and integrating them to provide simulation models of candidate systems.

1. INTRODUCTION

Typically, eighty percent of the cost of a product is in the components purchased to build it [1, 2]. (This is a *typical* figure, true of electronics, products made from motors, gears, drive amplifiers, etc. — even automobiles and aircraft. Obviously, it does not cover *all* products.) This estimate might even be low in certain fields, such as the personal computer business. Thus there is a crucial need to complement design

¹ Order of authors is alphabetical.

and manufacturing processes with a process of *component engineering*. The component engineering process brings concerns such as component performance, reliability, cost, and delivery schedule to the table — in time for these concerns to influence the performance, reliability, cost, and delivery schedule of the system that is being built from those components.

To address these concerns, Active Catalogs [14, 15] integrates solutions at a number of different levels. First, it provides a model of how to provide appropriate support for component engineering as an integral part of product development along with design and manufacturing. Second, it embodies a set of component capabilities that engineers need to perform component engineering. Third, it packages these capabilities in a way that is conducive to participation by large numbers of designers and manufacturers.

Active Catalogs helps designers find and evaluate candidate parts for components of their system designs. This general approach to component engineering is illustrated in Figure 1. The key concept is, “Try before you buy.” The goal is to help designers state component-level or system-level constraints that candidate parts should meet, find candidates, and then return information that helps designers evaluate how the candidates will work together in the system design. The information returned includes not just the static data found in catalogs today, but also behavioral information, including simulations and other active objects such as graphical animations. Extensions are also envisioned to provide information bearing on design for affordability and design for rapid production, such as cost, suppliers and other procurement data.

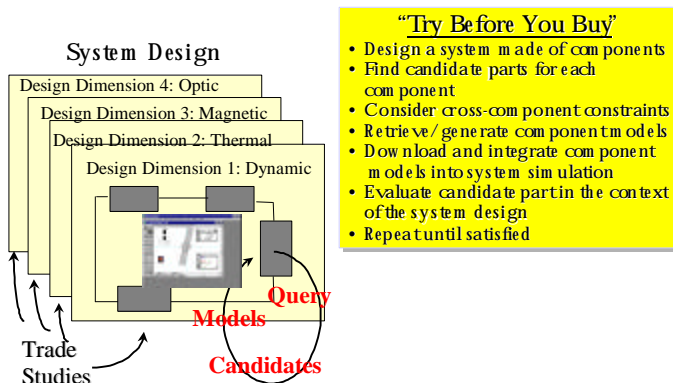


Figure 1: The Active Catalogs component engineering design cycle

Studies have shown that eighty percent of engineers’ time goes *not* into the design activities for which they were trained, but rather into assorted information management tasks [1].

Substantial parts of that time lie in planning and communications — much of it centered around obtaining information about components and evaluating tradeoffs with respect to design requirements. Active Catalogs’ contribution is to speed up and partially automate information collection and assessment activities such as development of simulations.

The basic concept is to provide, or augment, a workspace in which a designer is laying out the abstract components of a design. It augments the workspace by letting designers express constraints on individual or groups of components. On request, the collected constraints on a component can be folded into queries that retrieve candidates. For those candidates of interest to the designer, Active Catalogs retrieves technical specifications and models — or, in a number of cases, *actively generates* the models. The models can cover electrical, mechanical, kinematic, dynamic, thermal, magnetic, and optical modalities.

Designers can use these specifications and models in simulation environments to evaluate candidates for individual components, or interactions between multiple candidates. As they get better insight into their requirements from these evaluations, they can iterate the process to refine the constraints and narrow the set of candidates. When they are satisfied, the result is a bill of materials (and potential substitutes) that is known to fit the requirements — along with automatically recorded documentation of many of the factors that went into selecting the components.

Recently, Active Catalogs has been demonstrating its work by supporting gimbal design² with data on motors, encoders, actuators, tachometers and bearings. In addition to demonstrating the conceptual approach to component engineering in the context of gimbal design, we also report progress on technical capabilities pertaining to constraint-based specification of parts requirements for retrieval purposes, and on generating and presenting active objects such as animation models and simulations.

Active Catalogs is trying to build its software with clean interfaces that can work with arbitrary design systems. The approach to doing so is discussed briefly in Section 4.

2. CURRENT APPROACHES TO PROVIDING PARTS INFORMATION

The past couple of years have seen increasing interest in providing on-line catalogs to designers and engineers via the Internet and the World Wide Web. Currently, engineers can find on-line catalog data in three different forms:

² Gimbals are two-degree-of-freedom mechanisms used in tracking objects in guidance and control systems in the aerospace industry.

- 1) *General Web search services.* Engineers often use general search engine web sites, such as Yahoo [3] and InfoSeek [4], trying to find the parts they need by using the services of these web sites that specialize in locating companies. The user types in a list of keywords describing his search requirements. The search engine returns a list of part manufacturers' web sites that might contain technical information on the parts. Our experience with these search engine web sites is that they usually do not provide specific or comprehensive pointers that meet engineers' needs. The number of company web pointers is usually less than the number available from dedicated part catalog services, and many company web pages provide no technical part information, cost, or delivery data.
- 2) *Dedicated parts catalog referral services.* These services are dedicated to finding companies that can provide parts for engineers. Thomas Register [5] is one example of this type of service. As with general search engine web sites, the engineer types a list of keywords into Thomas Register and Thomas Register returns a list of company web sites that contain the technical part information. Compared to general web search services, Thomas Register provides a comprehensive list of companies — but not part data. The list of companies is specific and relevant to the needs of engineers, but the quality of technical information in the referred company web sites varies. Some of them have technical data and scanned drawings, while others provide only marketing and sales pitches.
- 3) *Centralized part catalogs.* Instead of providing a referral service to part manufacturer web pages, this service centralizes all the technical part information together at a single site. Users go to the site directly to select and purchase parts. PartNet [6] and IndustryNet³ are examples of this type of service. The technical information on parts is more organized and uniformly presented than individual company web pages. However, because the data is restricted to parts from manufacturers that have joined the service, the choice of manufacturers and parts is limited. Some manufacturers may be reluctant to participate in this service and would like to retain control of their product information themselves.

Existing catalog services have certain limitations in search capabilities and product information. They rely heavily on keyword-based string match, and their product information is limited to pictures and text descriptions.

To improve their services, catalog companies have been adding new search features. For example, in PartNet, part information is indexed in categories/subcategories, and users can use the taxonomy to find a particular part type. Users can specify attribute constraints for the selected part type. Also, Saqqara's [7] *step-search* provides an interface that displays a range of values for database attributes for user input, preventing invalid inputs. Users can select options one at a time, using the interface. When a reasonable number of candidates remain, users can invoke a comparison table for comparing the candidates. Centor Corp.'s [8] *parametric search* also provides valid inputs for attributes, and lets users choose several commonly used attributes for search. However, in these catalogs, users cannot describe requirements that are not explicitly addressed in the database. For example, it is not easy to describe implicit requirements such as product applications, the function of a product, or the environment in which a product will be used.

Some catalog providers offer a greater range of on-line information. In addition to text, images, and diagrams, they also provide information that helps users evaluate candidates. Micromo's [9] web site provides supplementary descriptions, including application notes, tutorials, and related links. However, the information is static in the sense that users have to read the given attribute values and the drawings and evaluate whether they satisfy their requirements. The Landrover [10] and BMW [11] web sites give a better way of evaluating products. They allow a dynamic environment for selecting vehicle models, colors, and desired features. Users can visually explore different combinations of options, before selecting a particular combination. However, exploration is limited to combinations of static visual features in a two-dimensional picture of one particular vehicle type. This is not the same as exploring dynamic performance.

3. ACTIVE CATALOGS SERVICES SUPPORTING COMPONENT ENGINEERING

The Active Catalogs system provides several services to support the component engineering process. These services enable a designer not only to find candidate parts to serve as individual components in a design, but to explore and assess interactions between candidate parts for sets of components. These services include mechanisms for: (1) creating queries for parts based on their intended use rather than merely parametric specifications; (2) refining those queries to take account of constraints imposed by other components; (3) providing multi-modal information to help designers assess and compare candidate parts; and (4) generating simulation models of candidate parts and integrating them to provide simulation models of candidate systems.

³ IndustryNet, a subsidiary of Parts, Inc., closed operations this year, but has been purchased by Perot Data Systems and may reopen or be reconstituted at a future time.

3.1. Ontology-Based Product Search Capabilities

Our work on retrieval is based on a domain theory (sometimes also called an ontology), which is used to map user requirements into database queries. Technical challenges being addressed concern the form in which these constraints are expressed, mechanisms for inferring constraints, interfaces to help designers express constraints, and methods for translating the constraints into productive queries on sources of parts information.

The retrieved information includes not only standard catalog data, but also multi-modality information, such as CAD and geometry files, and active objects such as animations and simulations. The challenge problems in this part of the effort are (1) *generating* active objects, such as simulation models for parts that provide only static data; and (2) *combining* multiple objects to support system-level evaluation, e.g., wrapping multiple component simulations to make it possible to compose them together to simulate a system.

The search system in Active Catalog maps high-level, conceptual models of products that are needed by customers into physical database queries. That is, it performs an inference of what attributes in what tables in the database are relevant to an implicit requirement that is given by a user. Also, when a user provides multiple different types of requirements, the system easily combines the given requirements together to produce a query.

To perform these functions, our ontology system provides several sub-features.

- **Mapping.** Each ‘requirement’ concept in our ontology is defined in such a way that it can be mapped into a particular product type and/or a set of attribute constraints. For example, a requirement about a motor function may be translated into a particular type of motors that can achieve that function.
- **Combining.** There are multiple sub-ontologies for multiple aspects of products. Using these ontologies, the user can create queries that combine different types of requirements.
- **Joining.** Multiple requirements can be combined by first applying the first sub-feature – the mapping – to the requirements, and then joining the mapping results through logical conjunctions. For example, if a user’s requirements consist of a product type, the application of the product, and the environment in which the product should be used, these three different requirements are combined through logical conjunction of the three mappings.
- **Translating.** We provide a translation table that maps the intermediate results (specific part type and attribute constraints) into database tables and their attribute constraints. The details of these sub-features are described below with examples.

3.1.1. Mapping a User Requirement into Part Types and Attribute Constraints.

The following procedure maps a user requirement into specific product types and/or attribute constraints. The output of the procedure (product type and attribute constraints) is mapped into database tables and database attribute constraints, as will be explained in the subsections following.

The mapping is performed in the context of finding a general product type, such as motors. If the given requirement is already about product type or attribute constraints, then there is no need for a translation, and the procedure passes it to the next step without change. Otherwise, the procedure applies a chain of rules to retrieve the implications of the requirement until it finds product types and/or attribute constraints implied from the requirement. This relies on powerful Loom [12] reasoning based on domain knowledge. For example, when the user needs electric motors *for high oxygen level environment*, the inference rules/constraints used are: (1) *high level oxygen environment is explosive*; (2) *brush incurs spark in the explosive environment and it is not safe*; and (3) *brushless motors do not have brush*. The mapping function returns *brushless motors*. Figure 2 shows the sub-network in our ontology that supports the above reasoning. The right-hand side hierarchy shows that *high oxygen level environment* is an *explosive environment* (1). The arrow from *explosive environment* to *brush motors* together with the connection between *Generate_Fire* and *Products* illustrates step (2). Finally, the definition of Brushless motors illustrates step (3).

Mapping (requirement, product type)

If the requirement is about product type or attribute constraints use the requirement description without change
 Otherwise, apply chains of rules and constraints to reach the set of specialized product types and/or the attribute constraints that satisfy the requirement

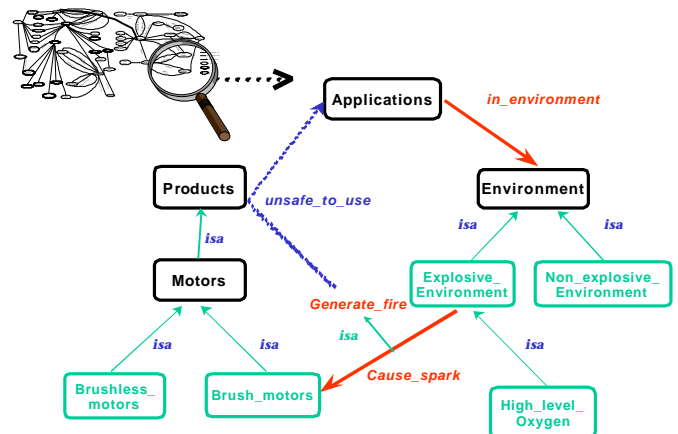


Figure 2: Finding motors for *high oxygen level environment*

3.1.2. Combining Multiple Source Requirements. We support multiple ways of describing products by defining multiple sub-ontologies for different aspects of user requirements. They are connected to the product taxonomy and attribute taxonomy directly or indirectly. Figure 3 shows a part of these additional sub-ontologies together with product taxonomy and attribute taxonomy. We have built taxonomies of electro-mechanical components, including an extensive hierarchy of various motors and pumps and a taxonomy of the attributes of these products. Currently, we have 300 classes of pumps, 3,000 attributes for pumps [13, 14], 72 classes of motors, and 66 attributes for motors.

We provide a semantic network of applications, intertwined with product and attribute taxonomies. As described earlier, each application can be mapped to a set of product types in a search context. In addition to that, given a specific product, the user may see the potential application of the product, following the link between product types and applications [15]. If a past application is available in the database, users can refer those details as well. Also, our function ontology classifies the products in terms of their generic functions. As shown in Figure 3, *amplify power*, *amplify voltage*, and *amplify current* are subclasses of class *amplify*, and these classes can be used for finding different kinds of amplifiers.

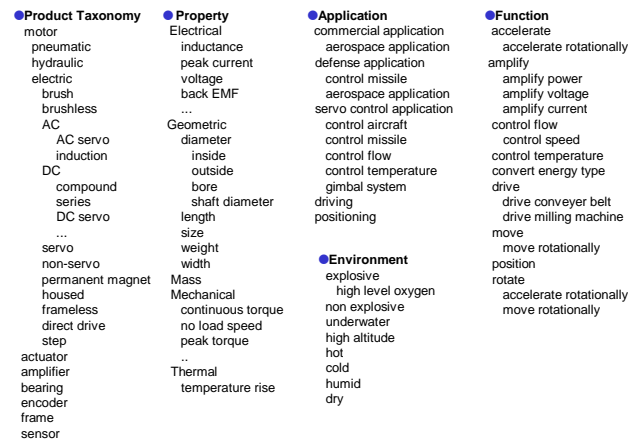


Figure 3: Sub-parts of the Active Catalogs ontology

In addition to these ontologies, we are planning to build a structure ontology and procurement ontology. Structural requirements include simple form features of products and topological arrangement of subparts. These requirements may be used in searches using the structure ontology. For example, the output interface of a motor can be either a shaft or, in the case of motors used in gimbals, a cavity in which other

components are placed. Depending on the interface type, the system can filter out irrelevant parts. Also, a motor may have a rotor, a stator, and a brush ring as its subparts. They may be fixed in a single unit or may be demountable, as required in gimbal design. The way a motor is constructed can provide clues that allow the system to find it. In the procurement ontology, we will describe product status, supplier track record, lead time, expected obsolescence date, etc., as well as supplier and cost. Once this portion is built, users might be able to examine this information before they buy, given appropriate authorizations from the suppliers.

Using these ontologies, users can describe their applications, functions, environmental restrictions, structures, and procurement requirements. Active Catalog combines them using the following procedure.

```

Combine_Requirements
  Compute implications of each requirement or a set of requirements
  together, using Mapping,
  to collect product types and attribute constraints, and
  combine them using logical conjunction 'and'
  
```

The above procedure first computes the implications of the given requirements, either separately, if they are irrelevant, or together, if they are related. Then it collects product types and attribute constraints from the implications. Finally, these types and attributes are combined through logical conjunctions to produce specific product types and their attribute constraints. For example, if there are two mapping results, *brushless motor type* and *DC motor type*, then they are combined into the intersection of the two types – *DC brushless motor* – which is also the common subclass of the two in this case.

We are planning to build a cross-constraint ontology that will allow constraint checking across multiple part types instead of one part type. This will allow users to check compatibility when they build a system from multiple components. When a user selects a particular type of part and then tries to select another that should be linked to the selected part, the system will check cross-product constraints to filter out inconsistent combinations. For example, if the user selects *DC motors* and then would like to select an amplifier for them, the system will recommend DC amplifiers, and remove AC amplifiers from the list of candidates, with the user's consent. This can be achieved by defining a relation, called *compatibility* in our knowledge base. The compatibility relation across different electro-mechanical product types is a sequence of constraint rules. Given two selected products (or product types), the system can check the compatibility between them by examining the sequence of constraint rules. If all of the rules are satisfied, the products are compatible.

3.1.3. Modifying Constraints Through User Interaction.

Our target user population, engineers, are expected to be familiar with normal “Netscape” style editing windows rather than with entering Lisp-like Loom expressions. We have provided the user with an editing-style interface in which the system presents the results of the mappings described in Sections 3.1.1 and 3.1.2. The user can modify or specialize the attribute constraints using the given editors, as shown in ‘Search Criteria’ frame in Figure 4. The construction of the user interface is driven from product types and attribute types to be edited and be displayed. That is, different data editors and display functions are invoked, depending on the attribute types.

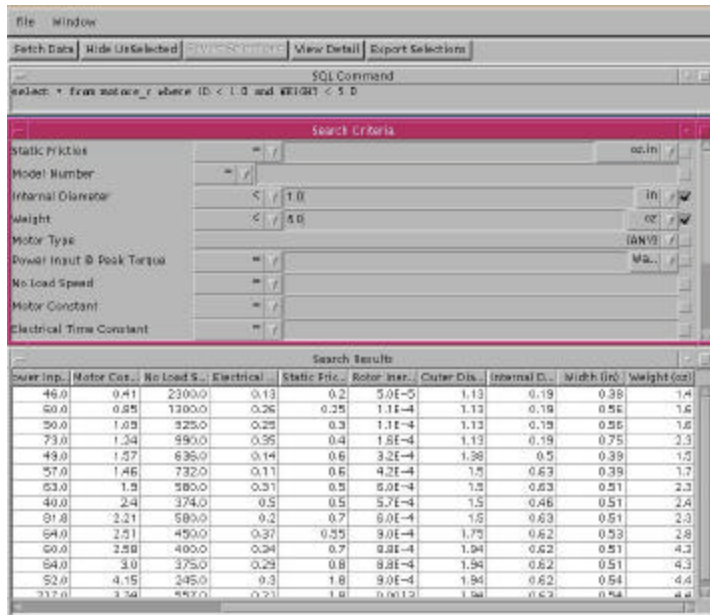


Figure 4: User interface for modifying attribute constraints and retrieving database results

First, the data types of the attributes determine the appropriate interfaces for describing search requirements and for displaying search results. For example, the Numeric Editor is used for attributes with numeric values. When users specify the search requirements, the Numeric Editor allows them to specify a range of values, such as *greater than X*, *less than Y*, and *between A and B*. That is, our interface changes depending on the ways users describe their requirements. If a user selects *between*, the system will provide two input fields instead of one. The String Editor is provided for string inputs, and the Selection Editor is provided for menu-driven inputs.

The Unit Editor allows the user to select different metric systems, depending on the attribute types and user preferences.

That is, the editor retrieves attribute properties in the Active Catalog knowledge base to support the appropriate interface for different attributes. For example, *weight* is represented in grams, ounces, and pounds; *length* is in meters, inches, feet, and miles; *temperature* is in °C, °F, °K; and so on. A translator is provided to change the user-selected units into the units in the database.

We are planning to subclass these editors depending on the product and attribute types. For example, Weight Editor and Length Editor will be instances of Numeric Editor. Also, their Unit Editors can be further specialized based on product types. For example, the Length Editor for Motors will have a Unit Editor that supports inches, feet, centimeters, and meters, but not miles or kilometers.

We have found that individual engineers often use a particular subset of attributes in searches. We support the use of user-defined subsets.

3.2. Creating Database Queries and Retrieving Results

The last step is the translation of product type and user attribute constraints into a database query. To perform this sub-function, we built a translation table (called meta-description). This table contains mappings between product types and database tables. It also contains mappings between general attribute names used by users and the attribute names in the database tables, depending on the names used by the data providers. The issue this addresses is that different manufacturers may use different names for what, conceptually, is really the same attribute. This mechanism provides mappings between a representative name preferred by the user and the product’s other names. Using this information, a database query is created from product type and attribute constraints. This process creates SQL queries to the Oracle database used by the Active Catalog system. A typical SQL query is ‘select * from TableName where Constraints’, where TableName is the translated database table name and Constraints is the combination of constraints translated through conjunctions.

As shown in Figure 4, the results are displayed in the ‘Search Results’ frame. The table in this frame lists all the candidates that satisfy all the requirements. The columns can be moved, so users can easily focus on the attributes they are interested in by modifying the column width. The user can also sort the candidates based on the values of a selected attribute. If the user selects a subset of candidates using the mouse, then the system hides the unselected candidates in order to highlight the selected ones. The ‘View Detail’ button invokes a display for displaying the details of the selected products, including images, application notes, and dynamic models. The details of this feature are explained in Section 3.3. Finally,

when the user is satisfied with a set of candidates, he or she can send the results to other system modules by pressing the 'Export Selections' button.

3.3. Multi-Modal Information

In addition to providing database information, Active Catalogs also provides part data in various modalities and data formats. For example, to determine the suitability of a servo control motor for a gimbal design, the engineer must assess its electrical requirements and mechanical performance, such as its power consumption and its peak torque. To determine whether a standard engineering part can actually meet the design requirements of a system, an engineer must evaluate the properties of the part in different dimensions, such as speed, torque rise time, and stability. However, the mechanical and electrical evaluations are not enough. Weight and volume are important: since a motor is mounted in a gimbal frame, the engineer must determine if the outside diameter of the inner gimbal motor can fit into the inside diameter of the outer gimbal motor. A motor also has its own weight; if it is to be mounted in an airplane or missile, the engineer must be sure that the weight of the motor does not violate its allocation in the total payload of the system.

While many of these properties can be viewed as database parameters and displayed in attribute/value pairs, engineers need other information and displays to give them a comprehensive view of the parts. Engineers would like to see a cross-sectional view or a two-dimensional drawing of the motor, showing the relationships of the rotor and stator within the motor as well as its inside and outside diameters. The relationships of the rotor and stator are not explicit in database tables. In drawings, engineers can determine the location of the mounting holes and the electrical terminal connections, information that is hard to present in a table. Other useful types of information include application notes describing how a part can be used, images showing various components of a part, and wiring diagrams showing the electrical connections and wiring, etc.

Figure 5: Display of multi-modal product information

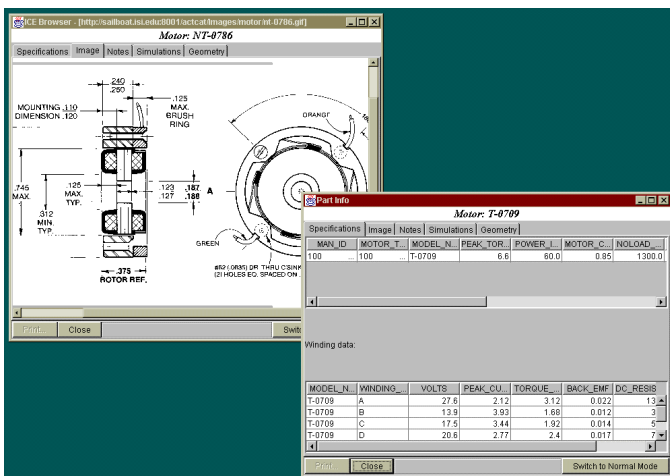
Most current on-line catalogs focus on providing technical parameters of parts in terms of database data. Engineers have to rely on other resources, such as talking to the part manufacturers directly, to obtain other information, such as drawings and application notes. Active Catalogs aims to provide all of the information engineers require in a cohesive, easily accessible, multi-modal form.

To provide this multi-modality capability, Active Catalogs supplements a regular database of parameter data with a special database to store the multi-modal part information. In addition, meta-data information is provided. Active Catalogs uses the meta-data to identify the relevant data formats and to provide appropriate displays. To provide an integrated display of multi-modal information, Active Catalogs uses a window frame, with separate folders for different types of modal information, to display individual parts. For example, Figure 5 shows two windows, each representing a different candidate part under consideration for use as a servo motor. In one window, the user has asked to see the motor's properties displayed in table format, including additional information about windings. In the other window, the user has asked to see the other motor displayed as a drawing. By clicking different buttons, engineers can switch between the different types of information available, and easily compare information about alternative parts. If they want to compare the degree of difficulty in the installation of two motors, they can look at their drawings and the application notes by putting the two windows for these motors side by side.

3.4 Active Models, Retrieved or Generated

In addition to providing multi-modal static information, we also generate and provide active and executable simulation models of various modalities. While multi-modal technical information such as text, database data, images and drawing files is useful, engineers need dynamic information to assess the performance of parts. For example, engineers need to know how fast a motor can rotate to a new equilibrium position when it is subjected to an input signal. One way to find this information is by applying a physical signal to the motor and observing the motor's behavior. However, physical testing of a motor is costly. In many cases, motors may not be available physically for testing. The practical way is to simulate the behavior of the motor in a simulation program. However, developing simulation programs requires programming expertise, and it is a tedious and time-consuming process.

Current catalogs, whether on-line or paper, do not provide simulation models. A key feature of Active Catalogs is its ability to provide simulation models, either hand-coded or automatically instantiated. Active Catalogs generates simulation models "on the fly" from database parameters. Motors have a well-known theory of operation, and the



parameters used in manufacturers' catalogs generally conform to it. The motor theory has rotor resistance and inductance, a back EMF coefficient, a constant velocity output torque to input current, an inertia and friction, etc. This information, combined with the load torque and inertia, is enough to generate differential equations and laPlace transform of a motor plus load. We have built generic models that are used with the parameters from the manufacturer to produce Matthews models automatically.

After a model is selected, Active Catalogs can launch that model with appropriate simulators, and display its behavior. Figure 6 illustrates this point. It shows a Matlab S-plane dynamic simulation model of a servo control motor and a VRML animation model that breaks apart the sub-components of that same motor.

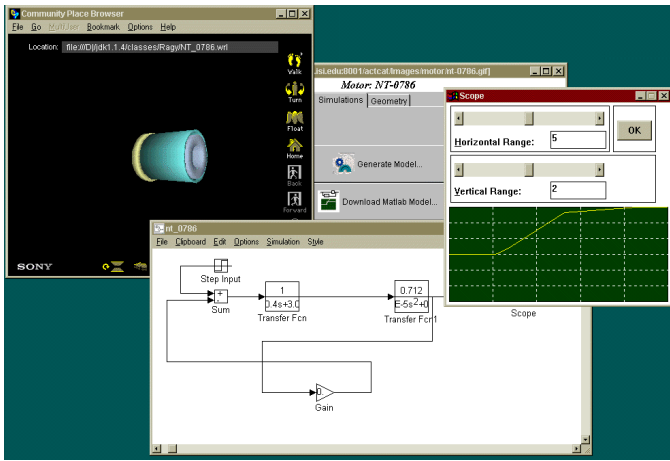


Figure 6: Active Catalogs simulation interface

Active Catalogs provides models in multiple modalities. Besides control simulation models, Active Catalogs can provide VRML animation models, showing the spatial and geometry aspects of a motor. These help engineers understand spatial and configuration aspects of parts within a system. Other model types include kinematics and dynamic models in Working Model. By providing a range of models, Active Catalogs gives engineers an environment to explore a part as if the part was physically in front of them.

In Active Catalogs, a generic model for a class of parts and queries is manually created and stored as a template file. This model is derived from the theory of operations mentioned above. Then the system creates, for example, a Simulink transfer function of a D.C. servo motor for computing the angular rotation of the motor in response to an input voltage signal. The model template file is expressed as a Matlab/Simulink model. The parameters of resistance, inductance, torque, etc., are symbolic values in the template file. When the model of a particular part is needed, Active

Catalogs retrieves the parameter values from the database, substitutes the symbols in the model template file with the database data, and then outputs the file with instantiated values as the specific model of that part. The process of automatically instantiating models for a specific part from a generic model is shown in Figure 7. The technology for the generation of these models derives from the now almost lost art of programming analog computers.

4. Future Directions

The Active Catalogs effort is developing a general architecture and model of operations adaptable to a wide range of applications that need component engineering. Examples include unmanned aerial vehicles, affordable missile systems, optical/magnetic/electro/mechanical systems, and advanced sensors. This vision cannot be realized without access to large bodies of basic component information. Our future work lies in facilitating the creation and consumption of active information, particularly the models and meta-data needed to bring component engineering into the design process.

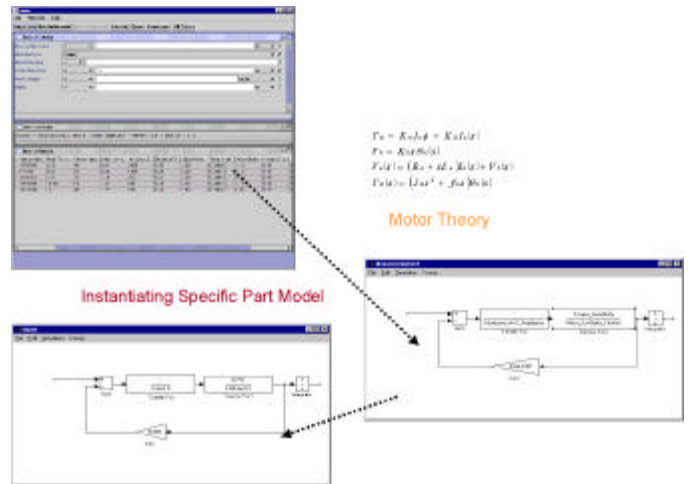


Figure 7: Automated instantiation of a specific part model

One of the key challenges in scaling up the system is to provide a framework that enables distributed participation by users, product information suppliers, and providers of modeling services. The framework must also support distributed participation within an organization, recognizing that, many times, design evaluations are performed by different engineers with specialized expertise. Furthermore, the framework needs to be designed with an eye to simplifying system maintenance and upgrade functions.

Our approach has been to move toward a replicated, decentralized client-server architecture. One of the key ideas behind the architecture is a network of communications both

between local servers, and between servers and their clients. At each site, a local server provides in-house access control to proprietary product data, modeling, and analysis services. The local server also filters access to external data and services, reflecting organizational concerns such as preferred providers and debarred vendors. Local servers also control access from other, external servers, determining under what circumstances data and services at that site will be made available to others.

The intent behind this scheme is to provide the high degree of organization-specific control and security needed for comfort by engineering organizations, while simplifying insertion of new product information and modeling services. Providers will register with any local server, and have their offering disseminated throughout the network in a manner similar to the process by which updated information about new sites disseminates through the Internet.

The client-side architecture for Active Catalogs to which we are moving focuses upon supporting assessment of designs with respect to multiple modeling views: control, kinematic, electrical, thermal, and others, including the ability to create hybrid models (e.g., linking a control and a kinematic model).

Two key considerations drive this part of the architecture. The first is to establish a very clean separation of user interface and underlying application services in the client, with externally accessible application-programming interfaces for every service. This is intended to allow Active Catalogs to operate in either stand-alone mode or in conjunction with an organization's pre-existing suite of CAD tools. (For example, we are currently engaged in interfacing Active Catalogs to Lockheed-Martin's experimental Integrated Gimbal Design system in a joint experiment supported by the Defense Advanced Research Projects Agency.) The second consideration is to provide support for a complex constraint web that coordinates participation in multiple modeling views of the objects that represent components and candidate parts for those components.

In addition to these issues of engineering for adoptability, we are exploring several long-term issues:

- Lowering participation costs via tools that help manufacturers convert to active versions of their regular catalogs.
- Improving designers' ability to assess candidate parts by supporting model interoperability across domains, such as thermal, stress, optic, and magnetic.
- Facilitating participation from design environment providers through tools and protocols for customizing simulation models for a design environment.
- Interfaces that improve productivity by facilitating search by function, geometry, structure, and application.

- Security methodology to protect manufacturers' intellectual property and to prevent back-engineering while allowing and encouraging the provision of accurate behavioral models to the user.

REFERENCES

- [1] P. Will, "Simulation and Modeling in Early Concept Design: An Industrial Perspective," *Research in Engineering Design* (1991) 3:1-13
- [2] Committee to Study Information Technology and Manufacturing, P. Will, chair, *Information Technology for Manufacturing: A Research Agenda*, National Academy Press, 1995.
- [3] Yahoo. <http://www.yahoo.com/> 2/2/1998.
- [4] Infoseek. <http://www.infoseek.com> 2/2/1998.
- [5] Thomas Register <http://www.thomasregister.com/> 2/2/1998.
- [6] PartNet. <http://www.partnet.com>, 1/30/1998.
- [7] Saqqara. <http://www.saqqara.com>, 1/30/1998.
- [8] Centor. <http://www.centor.com>, 1/30/1998.
- [9] Micromo. <http://www.micromo.com>, 1/30/1998.
- [10] Landrover. <http://www.landrover.com>, 1/30/1998.
- [11] BMW. <http://www.bmwusa.com>, 1/30/1998.
- [12] R. MacGregor, *Loom User Manual*, USC/Information Sciences Institute. Working paper ISI/WP-22, 1990.
- [13] P. Luo and P. Will, Active Catalog: A Knowledge-Rich Design Library Facilitating Information Consumption, in *ISIP Workshop Series on Knowledge Intensive CAD-2*, 1996.
- [14] S.R. Ling, J. Kim, P. Will, and P. Luo, Active Catalog: Searching and Using Catalog Information in Internet-Based Design, in *Proceedings of DETC '97: 1997 ASME Design Engineering Technical Conferences*, Sacramento, California, September 14-17, 1997.
- [15] J. Kim, S.R. Ling, and P. Will, Ontology Engineering for Active Catalog, in *AAAI Workshop on Using AI in Electronic Commerce, Virtual Organizations and Enterprise Knowledge Management to Reengineer the Corporation*, 1997.