Smart Catalogs and Virtual Catalogs

Arthur M. Keller
Stanford University
Computer Science Dept.
Stanford, CA 94305 USA
ark@cs.stanford.edu

Abstract. We present an architecture for electronic catalogs, called Smart Catalogs and Virtual Catalogs. Smart catalogs are searchable, annotated combinations of machine-readable (i.e., minimally processable) and machine-sensible (i.e., actually understood by the computer) product data. Virtual catalogs dynamically retrieve information from multiple smart catalogs and present this product data in a unified manner with its own look and feel, not that of the source smart catalogs. These virtual catalogs do not store product data from smart catalogs directly (except when caching for performance); instead virtual catalogs obtain current product data from smart catalogs to satisfy specific customer queries. Customers interact with smart catalogs and virtual catalogs through WWW or other interfaces.

1. Introduction.

Electronic catalogs are a key component of electronic commerce. The procurement process extends from product selection to source selection to negotiation of price and other terms and conditions to ordering to order fulfillment to payment. Electronic catalogs are the reference for product selection and can assist with source selection and description of terms and conditions. Other electronic commerce components, such as EDI, handle the other aspects of the procurement process.

Electronic catalogs can be organized as individual company catalogs or they can participate in a multi-catalog framework. Currently, companies that have electronic catalogs organize them uniquely. Companies perceive a competitive advantage from how they organize their catalogs. Furthermore, each company has to base its electronic catalog at least in part on legacy systems or legacy organizational structure, and these influence the nature of the electronic catalog produced.

The challenge is to enable companies to have their catalogs to participate in a multi-catalog framework while still retaining the uniqueness of the catalogs. That is, these catalogs are highly heterogeneous. Often the catalogs are organized as a collection of static HTML (HyperText Markup Language) documents for the WWW (World-Wide Web). Sometimes these catalogs are stored in databases.

Electronic catalogs organized as a collection of static HTML documents typically use ordinary WWW protocols and WWW clients, such as Mosaic or Netscape. Each HTML document, called a page, is stored in a separate file and these documents can link to other documents somewhere on the Internet. You can think of the World-Wide Web as a giant menu system, where each document contains content as well as links to other documents. Typically there are no roadmaps to the WWW, so the user navigates from document to document, hopefully getting closer to the information desired.

Each vendor maintains its own (collection of) catalogs. There may not be uniformity even among multiple divisions of the same company, let alone among multiple companies manufacturing or selling comparable products. Because of the difficulty of locating documents within a catalog, some vendors provide search interfaces for their own catalogs. Typically, these search interfaces

are for keyword or text search, although some sites support structured searches of databases.

There are a variety of limitations of this typical approach to electronic catalogs. It is hard to find what you are looking for. You have to figure out where to start. For example, in order to find information about a particular product, you must first determine which companies make that product and then separately visit each company's catalog, and then navigate through each of those catalogs separately to find the desired product. Making such navigation more difficult is that each vendor organizes its catalogs differently.

Few catalogs have the ability to search by content (i.e., reverse-search). Those that support reverse-search typically use keyword- or text-based searches. Keyword searching techniques, such as WAIS, are only of limited help, as they require that the user phrase the query using the terminology of the each data source, and vendors tend to use differing terminology from each other to describe products. For example, consider a search for VCRs with editing capability. Different manufacturers use their own proprietary names to describe editing capability. The user may not know the distinctive name used by each manufacturer. Also, keyword searches typically cannot be used to answer such queries as VCRs without editing capability.

Some electronic catalog systems, such as Step Search by Saqqara Systems, support structured search. However, this approach currently only works for a single catalog at a time. PartNet is experimenting with structured cross-catalog search. However, PartNet requires that the catalogs be stored in relational databases and has only limited support for heterogeneity. PartNet supports different database systems, different field and table names, and some unit conversion, but not more complex translations among heterogeneous catalogs.

In contrast, our approach is to support reverse-search (i.e., search by content) of multiple vendor catalogs based on a deeper understanding of the contents of these catalogs, so it supports an extensive framework for heterogeneity.

Section 2 describes the goals and principles of Smart Catalogs. Section 3 describes the overall architecture of a Smart Catalog. Section 4 describes the concept of a Virtual Catalog. Section 5 contains a usage scenario of Smart Catalogs and Virtual Catalogs. Section 6 describes the current status and future work of this project. Section 7 gives our conclusions.

2. Goals and Principles.

The primary goal of the Smart Catalog approach is to enable the creation of single company catalogs with powerful search mechanisms that facilitate the transition to multi-company cross-catalog search mechanisms. A secondary goal is to enable the reuse of catalog information for other purposes than originally intended.

In May 1994, the CommerceNet Catalog Working Group developed a list of about 30 requirements for electronic catalogs. These requirements can be summarized as follows. The system should be scalable and support distributed search (not centralized). Heterogeneous information sources should be supported (not one format or one structure forced on all catalogs). The system should provide up-to-date information (printed catalogs and CD ROMs are obsolete as soon as they are produced). The system should have an open architecture (allow connection of new information sources, adhere to standards, and allow integration of other approaches and legacy systems). A variety of search techniques should be supported (e.g., menus, keyword, text-based, parameterized, structured "reverse-search"). Cross-search of multiple catalogs for comparable products should also be supported by each catalog.

The requirement for an open architecture supporting a variety of search engines led to several design decisions. First, we the query processing core from the data sources and the user interfaces. Thus, multiple user interfaces can access the remainder of the system. Also, a variety of data sources can be connected including various search engines. Secondly, we use an intelligent-agent-based architecture. This architecture supports such functions as translation, routing, and notification. Thirdly, internally we use a very powerful query and data description language that is capable of encoding practically any data source or query capability. This language is KIF (Knowledge Interchange Format). KIF was developed as part of the ARPA Knowledge Sharing Initiative and is currently undergoing standardization by ANSI. We also use KQML (Knowledge Query and Manipulation Language) in order to communicate KIF among the agents in our architecture. KQML was also developed for the ARPA Knowledge Sharing Initiative. We use ontologies to formally describe the structure and the terms of each catalog as well as the relationships among these terms.

3. Architecture.

Our architecture is illustrated in Figure 1. It consists of intelligent agents communicating with facilitators, plus other components, such as data sources and user interfaces, that interact with the collection of intelligent agents and facilitators.

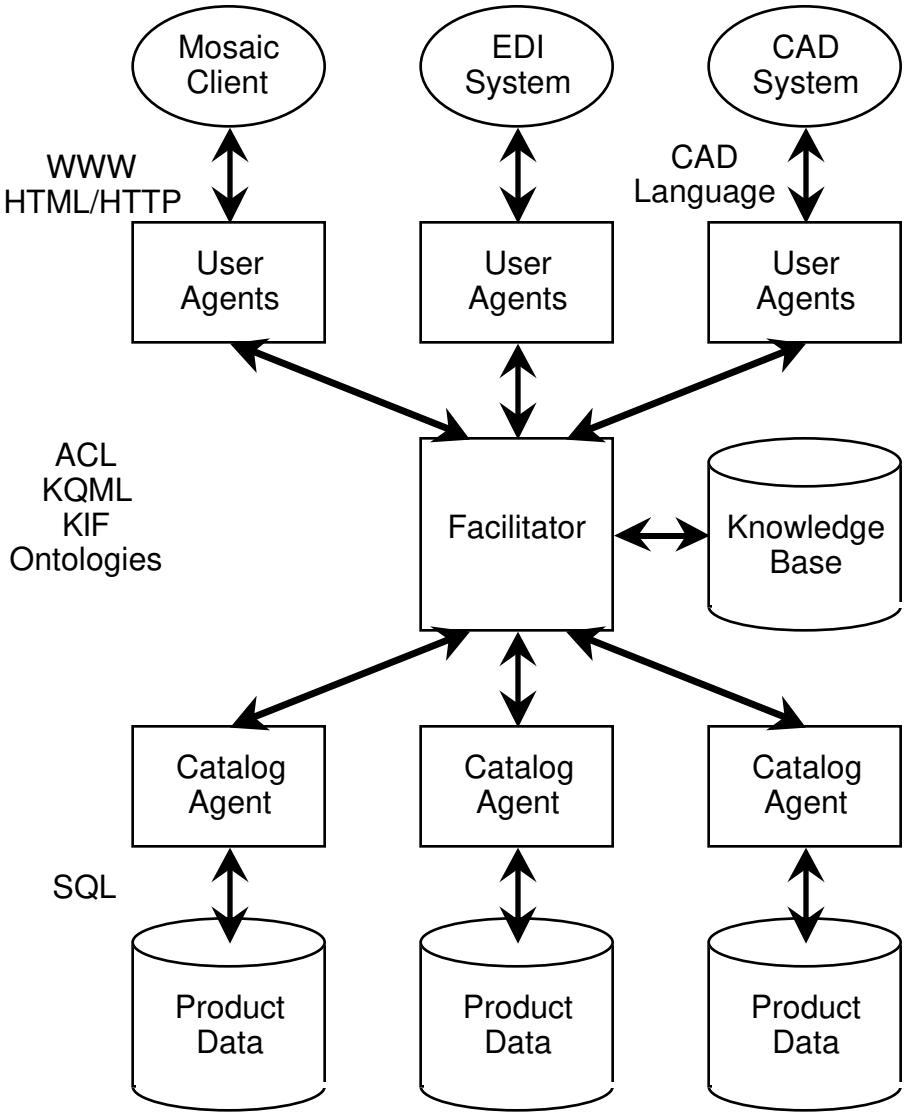

Figure 1. Smart Catalog and Brokering Architecture for Electronic Commerce

The communication language used by the intelligent agents in our architecture is Agent Communication Language (*ACL*). ACL consists of the Knowledge Query and Manipulation Language (KQML), the Knowledge Interchange Format (KIF), and a set of Ontologies. *KQML* consists of performatives, such as `ask-one`, `ask-all`, and `tell`, that describe the nature of the action to be taken. KQML has the role of the communication language in CORBA. *KIF* is based on First-Order Predicate Calculus and is the content language we use with KQML. KIF is powerful enough to contain or to encapsulate any other first-order or simpler content language, so that any information may be obtained from the information source, translated to the desired format, and transmitted to the requester, assuming the necessary components exist. Each product database is described by an *Ontology* (another term for ontology is "controlled vocabulary"), which defines the database, its structure, the terms used in it, and how they relate to each other. You can think of KIF as the content language, KQML as the transport protocol, and ontologies as the terminology used within the KIF language for interoperating intelligent agents in our architecture.

Each Facilitator in our architecture acts as a broker. Facilitators perform routing and translation. Each facilitator stores agent-provided advertisements of coverage in a knowledge base along with relevant ontologies. For example, an agent may advertise that it can respond to queries for VCRs. The facilitator uses these advertisements to determine which agents can support a particular request. For example a request for Super-VHS VCRs with editing capability will be given to the above VCR agent among others. And the facilitator translates requests into the language and terminology used by each responding agent and also translates responses into the language and terminology used by the requesting agent. For example, one manufacturer refers to editing capability as "random assemble editing" while another manufacturer uses the terms "Control L" and "Control S" for their editing systems. A facilitator will decompose requests requiring action by multiple agents and then compose the responses for the requester. For example, a request for reliable used cars for sale for less than $5000 that get more than 30 miles per gallon will be decomposed into a request to a classified-advertisement service for used car listings, a request to a car-rating service for the ratings of those cars, and a request to a car-specification service for gas mileage for the reliable cars for sale.

Product data is stored in databases, for easy search and maintenance. Such data includes structured information, parameters, text, pictures, sound, video, etc. Each product database communicates with a Catalog Agent using its native language, such as SQL. The Catalog Agent performs 3 roles: It advertises the coverage of a product database; it understands queries and translates them into the language of the product database, and it packages answers from the product database into ACL, our standard format for communicating among intelligent agents.

Consider the query for Super-VHS VCRs with editing capability. Support that there is a Catalog Agent that advertises to facilitators that it can respond to requests for VCRs. Its ontology defines the concept "Random Assemble Editing." Its database specifies which of the company's Super-VHS VCRs have Random Assemble Editing. The relevant facilitator has a taxonomy that indicates that Super-VHS VCR is a subclass of VHS VCR, which is in turn a subclass of VCR. The facilitator's knowledge base has a rule that Random Assemble Editing is a form of editing capability. Our query for Super-VHS VCRs with editing capability is translated by the rule to Random Assemble Editing. Because VCR is a superclass of Super-VHS VCR, our VCR agent is given the request for Super-VHS VCRs with Random Assemble Editing. The agent converts this ACL query into an SQL query against the catalog database. The result is packaged by the agent into ACL. Depending on the request of the user interface agent, the facilitator will return the resulting information about Super-VHS VCRs with Random Assemble Editing, or will translate these reponses into Super-VHS VCRs with editing capability.

Someone using a WWW client, such as Mosaic or Netscape, will connect to a User Agent using the ordinary WWW protocols HTTP and HTML. The User Agent will present the user with an HTML form, either statically or dynamically created. The user will describe the desired object using an HTML form. When responses come back from the facilitator, the User Agent will prepare dynamically created HTML documents with those responses.

We define four types of ontologies. *Base ontologies* are used to define common terms, such as engineering math, legal terminology, standard terms and conditions. Base ontologies are shared among all uses of this approach and are created by universities and research laboratories. For example, a common

business term for payment timing is "2 10 net 30," which means that a 2% discount may be taken for payment within 10 days, and full payment must be received within 30 days. It is unreasonable for each catalog using our approach to have to separately define this term, but rather such definitions should exist for all to use. *Domain ontologies* contain terms common to all or most vendors in an area, such as CPU speed, RAM size, or disk storage capacity. Typically domain ontologies are created by standards bodies and trade associations. *Product ontologies* contain company-specific terminology and refer to domain ontologies, such as NuBus cards for the Apple Macintosh. Individual companies create product ontologies, although other companies may refer to them. *Translation ontologies* are used to translate specific terms used in one ontology or information source to related terms used in another ontology or information source.[1] For example, a translation ontology may describe that Random Assemble Editing is a form of editing capability for VCRs. Individual companies create translation ontologies to enable them to compete in other markets. We expect there to be service organizations that create and maintain product ontologies and translation ontologies on behalf of other organizations.

4. Virtual Catalogs.

One important problem with using the WWW for product catalogs is the interaction between manufacturers and distributors or retailers. Consider a retailer that sells products from multiple manufacturers. The retailer will want to include product information from each manufacturer in its product catalog. Replicating all this product information in the retailers catalog would incur a considerable storage and maintenance cost. The typical current approach using the WWW is for the retailer to hyperlink to each manufacturer's catalog so that the customer may obtain detailed product specifications.

There are several problems with the hyperlink approach. First, the customer may get "lost" within the manufacturer's webspace and not know how to get back to the retailer. Second, the manufacturer does not know the context of the customer's interactions with the retailer. Third, the customer may stumble upon a how-to-order page provided by the manufacturer, and wind up ordering from someone other than the original retailer. Fourth, if the customer does make it back to the original retailer by using the "back" button, no information determined at the manufacturer's site is carried along with the customer, such as the desired product configuration. Fifth, if the customer gets back to the retailer through the manufacturer's how-to-order page, the retailer does not know the original context of the interaction with the customer (e.g., other products selected for order in this same session).

One approach to multivendor catalogs is the integrated approach, where all the catalogs are stored on one site using one implementation. A notable example of this approach is produced by Open Market. An alternative is to provide some mechanism for manufacturers to respond to specific queries by retailers to satisfy customer requests for product information. This latter approach can be based on a business relationship between the retailer and the manufacturer. That approach is the one we take in a Virtual Catalog.

Virtual catalogs allow retrieval of product data using a distributor's catalog by combining information from multiple manufacturers' catalogs. This retrieval is performed dynamically, upon the user's request, based on the user's search criteria, using the terminology of the distributor or any connected vendor, and will retrieve data from any relevant connected vendor. Therefore, the

---

[1] The term "articulation axiom" is sometimes used to refer to entries in a translation ontology. Translation ontologies may be used to create an ontology algebra supporting translation across multiple ontologies.

distributor's virtual catalog is always kept up-to-date and in synchrony with each manufacturer's smart catalog. The distributor can choose to display all of a manufacturer's products or only a subset of those products. Also, manufacturer's catalog information may be cached at the virtual catalog site and updated as it changes. Translation of terminology and concepts may occur when information is retrieved to be cached or when information is retrieved from the cache, or a combination of these times.

With virtual catalogs, the distributor maintains control over the interaction with the user. The user never interacts directly with the manufacturer's catalog. Instead, the manufacturer's information is retrieved on demand and is presented to the user with the distributor's look and feel, not the specific look and feel of each manufacturer. The relationship between the user, virtual catalog, and smart catalogs is shown in Figure 2.

There are two key business relationships in the Virtual Catalog world, and many supporting business relationships. The two key business relationships are between the customer and the retailer (the virtual catalog), and between the retailer and the manufacturer (the smart catalog). Processes may exist in the virtual catalog to bridge these relationships. For example, orders from customers may trigger resupply EDI transactions to the manufacturer. Or the order may be forwarded to the manufacturer for drop shipment of the product directly to the customer. In addition, there are supporting business relationships. For example, if credit cards are used for payment, the customer has a relationship with the issuing bank, and the retailer has a relationship with the acquiring bank, and the two banks have a relationship for clearing the credit card charge. Similarly, there are relationships for order fulfillment, shipment, etc.

Virtual catalogs are appropriate for retailers and distributions as well as in-house procurement catalogs, but they also enable new business models. A virtual distributor may operate using a virtual catalog and business relationships for order fulfillment, shipment, etc. The virtual distributor may not even have any inventory, warehouse, etc. The virtual distributor could be a completely computerized setup, automatically providing product information using manufacturers' catalogs, taking orders and arranging for order fulfillment and payment.

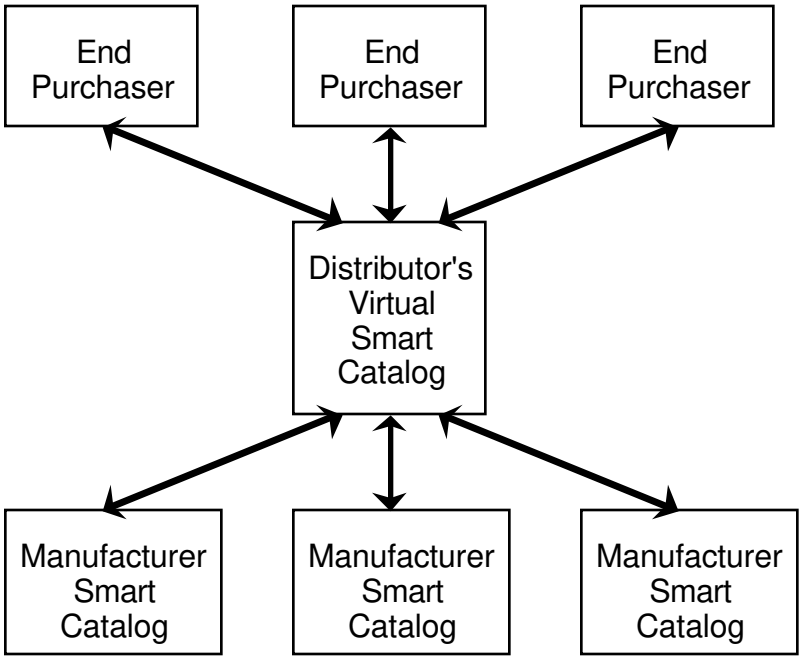

Figure 2. Virtual Catalogs.

5. Scenarios.

Consider a customer's request for color PostScript inkjet printers for the Macintosh costing under $1000. The User Agent will translate the query into KIF and submit it to a Facilitator. The Facilitator handling the query will consult its knowledge base for the facilitators or agents that can handle this request. For example, the Facilitator may transmit the request to the Catalog Agents for Apple and for Hewlett-Packard. The Catalog Agent will then interrogate the product database and translate the answer into KIF. For example, the Hewlett-Packard Catalog Agent may respond with the description of the HP DeskWriter 660C printer. The Apple Catalog Agent may respond with the Apple Color StyleWriter Pro. The facilitator will then collect these responses for the User Agent, which will package the responses in HTML for the Mosaic client. The user agent and facilitator are provided by the virtual catalog company. The catalog agents and product databases are provided by the manufacturers as part of their smart catalogs.

A customer may instead be interested in color PostScript laser printers for the Macintosh for under $3000. As of this publication, such printers cost around $5000, but prices are dropping. So the customer may request notification when any such printer is newly announced, or is lowered in price. This request will be stored in the relevant facilitator's knowledge base. When a manufacturer announces a new product, it will have a catalog agent send a message with the announcement and any changes to the advertisement of the agent to the facilitator. The facilitator will then send appropriate notifications to those parties who have expressed interest in this news.

Notice that the facilitator notification scheme is symmetric. Catalog agents express interest in being given product data queries. User agents express interest in being given product data announcements. The same notification scheme is used for both types of activities.

Operators of virtual catalogs have several alternative models for paying for their operation. The operator may take and process

orders and use the markup on the transaction. Alternatively, the operator may make money merely by providing information. (When information is free, search is valuable.) The operator may charge manufacturers, either a fixed fee or per referral. The operator may charge customers, either by subscription or per search. The operator may sell demographic information on customers or on searches to market research firms, manufacturers, retailers, or trade associations. Of course, an operator may obtain revenues from multiple of these approaches.

6. Current status.

The architecture of smart catalogs and virtual catalogs is an application of the facilitator architecture long in use in Logic Group of Stanford University's Computer Science Dept. The facilitator architecture has been demonstrated in the domains of software interoperability and concurrent engineering. It is now being applied to the domain of electronic commerce as part of Stanford's Center for Information Technology's (CIT) efforts on CommerceNet.

Several smart catalogs have been built in collaboration with several companies in the domains of workstations, test and measurement equipment, and semiconductors.

We have also created smart catalogs for several manufacturers and a virtual catalog that can search these smart catalogs. We are working on creating additional smart catalogs for a larger scale trial.

Please contact Arthur Keller by e-mail at `ark@cs.stanford.edu` or by WWW at `http://logic.stanford.edu/cnet.html` for more information.

7. Conclusion.

We have described an architecture for electronic catalogs for multiple companies that interoperate. Companies create smart catalogs of searchable, machine-sensible product information. Retailers and distributors create virtual catalogs that provide customers with product information dynamically requested from manufacturers' smart catalogs. Virtual catalogs provide a new degree of interaction between manufacturers and retailers or distributors. Virtual catalogs enable new business relations and new business models.

Acknowledgments.

This paper has benefited from feedback of numerous people who have heard presentations of this paper, read earlier drafts, or participated in discussions. In particular, Michael Genesereth designed the overall intelligent agent communication architecture used by smart catalogs and virtual catalogs, Narinder Singh implemented the facilitator, and Mustafa Syed programmed some of the other components in our initial prototypes. Note that an earlier version of this paper, co-authored by them, appeared at the Workshop on Electronic Commerce following CIKM, December 1994. A subsequent version of this paper appeared in the USENIX Workshop on Electronic Commerce, July 1995. Infomaster, a virtual information system representing our second generation smart catalog technology, was developed by Donald Geddis. Several other people have worked on the development of smart catalogs. These include Felix Chow, Rohan Aranha, Bob Engelmore, Wanda Pratt, and Rupert Brauch.

This work was funded through a subcontract from the CommerceNet Consortium, which in turn derived its funding from a cooperative agreement with the U.S. Technology Reinvestment Program, as well as over 150 companies and organizations. In addition, several companies have provided funding of this project towards the construction of pilot smart catalogs.

References.

ANSI X3T2, "Knowledge Interchange Format Reference Manual," Michael R. Genesereth, ed., March 1995, available from URL http://logic.stanford.edu/papers/kif.html.

Saqqara Systems' StepSearch can be seen at URL http://www.saqqara.com.

DARPA Knowledge Sharing Initiative External Interfaces Working Group, "Specification of the KQML Agent-Communication Language," Tim Finin and Jay Weber, eds., February 9, 1994, available from URL http://logic.stanford.edu/papers/kqml.html.

Michael R. Genesereth and Steven P. Ketchpel, "Software Agents," Comm. ACM, Vol. 37, No. 7, July 1994.

Thomas R. Gruber, "Toward principles for the design of ontologies used for knowledge sharing," in Nicola Guarino, Ed., International Workshop on Formal Ontology, Padova, Italy, 1992.

Open Market's URL is http://www.openmarket.com

William T. Wong and Arthur M. Keller, "Developing an Internet Presence with On-line Electronic Catalogs," Workshop on Electronic Commerce, December 1994, available from the URL http://www-db.stanford.edu/pub/keller/1994/cnet-online-cat.ps.