# A Smart Catalog and Brokering Architecture for Electronic Commerce

Arthur M. Keller
Michael R. Genesereth
Narinder P. Singh
Mustafa A. Syed

We present an architecture for smart catalogs and brokering that supports cross-search of multiple catalogs.  Typically, the World Wide Web (WWW) resembles a giant menu system, where each document behaves like a menu of links to other documents that get the user closer to the information desired.  For example, in order to find information about a particular product, you must separately visit each vendor of that product, and then navigate through that vendor's Web space to find the desired product.  Making such navigation more difficult is that each vendor organizes its Web space differently.  Keyword searching techniques, such as WAIS, are only of limited help, as they require that the user phrase the query in the terms of the information, and vendors tend to use different terminology from each other to describe products.  Our approach is to support reverse-search of multiple vendor catalogs based on a deeper understanding of the contents of these catalogs.

Our architecture is illustrated in Figure 1.  Product data is stored in databases, so that it may be readily searched and maintained.  Such data will include 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 in a standard format.

The communication language used by the 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 content language, so that any information may be obtained from the information source, translated to the desired format, and transmitted to the requestor, assuming the necessary components exist.  Each product database is described by an Ontology, which defines the database, its structure, the terms used in it, and how they relate to each other.  Base ontologies are used to define common terms that may be used by product ontologies.  Translation ontologies are used to translate specific terms used in one database to related terms used in another database.

A Facilitator in our architecture acts as a broker.  It stores agent-provided advertisements of coverage in a knowledge base along with relevant ontologies.  It uses to advertisements to determine which agents can support a particular request.  And it translates requests into the language and terms used by a responding agent and also translates responses into the language and terms used by the requesting agent.  A

Facilitator will decompose requests requiring action by multiple agents and then compose the responses for the requestor.

Someone using a WWW client, such as Mosaic, will connect to a User Agent using the ordinary WWW protocols HTTP and HTML.  The user will describe the desired object using an HTML form.  For example, the request may be to find color 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 560C 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.

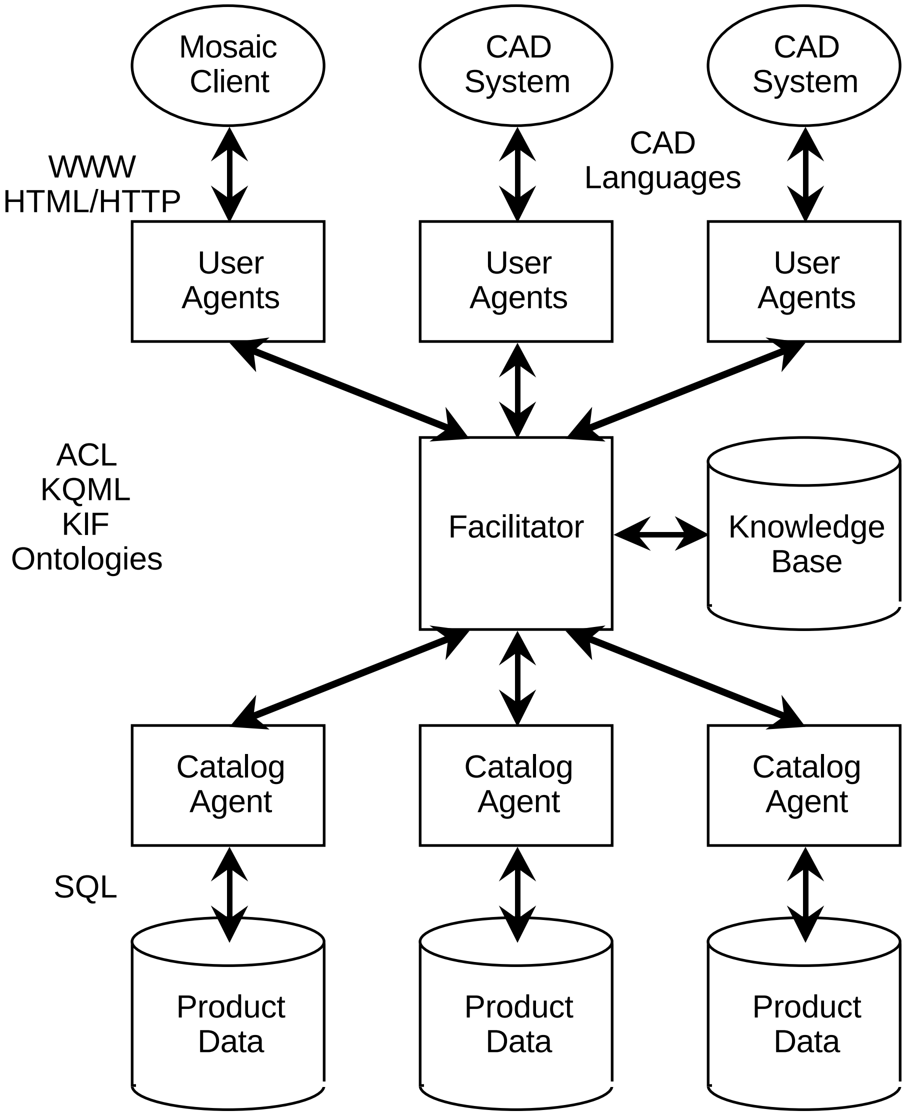

Figure 1.  Smart Catalog and Brokering Architecture for Electronic Commerce

This 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.  Please contact Arthur Keller at `ark@cs.stanford.edu` for more information.