Building Database-driven Electronic Catalogs

Sherif Danish

Saqqara Systems, Inc.
1230 Oakmead Parkway, Suite 314
Sunnyvale, CA 94086
408-738-4858
sdanish@saqqara.com
www.saqqara.com

Abstract: This paper describes issues and solutions related to the creation of a product information database in the enterprise, and using this database as a foundation for deploying an electronic catalog. Today, product information is typically managed in document composition systems and communicated on paper. In the new wired world, these processes are undertaking fundamental changes to cope with the time to market pressure and the need for accurate, complete, and structured presentation of product information. Electronic catalogs are the answer.

Introduction

Browsing a manufacturer's paper catalog is no longer good enough to make informed buying decisions. Paper catalogs take a long time to produce, print, and distribute. They are quickly obsolete, and they often do not contain enough information. The catalog production cycle is even longer for distributors because they have to wait for the manufacturer paper catalogs before they create their own.

Putting an electronic version of a paper catalog online does not solve the problem because it is difficult for the buyer to navigate and difficult for the seller to maintain. Online users do not want to browse catalog pages. They want to search and find quickly the product they are looking for.

The way to solve this problem is to structure product information and store it in a database. Structuring product information consists of classifying products into families, and entering the individual attributes of each product in separate fields in the database. Managing product information in a database allows product managers to apply updates directly into the database, and allows customers to perform accurate queries based on required product attributes. In addition, the database content can be transmitted directly to a distributor, thus creating a smooth propagation of product information throughout the supply chain.

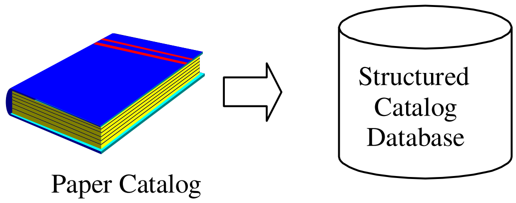

FIGURE 1. The first step is to convert printed catalogs into a structured product information database.

The following sections discuss issues and techniques related to the implementation of product information databases throughout the supply chain.

Starting Point: Component Manufacturers

Component manufacturers are the very first source of product information. They feed the databases of not only their distributors, but also the databases of the next manufacturers in the supply chain. Integrated Circuit (IC) manufacturers for example provide their product information to computer manufacturers. The catalog of the computer manufacturer is expected to provide, for example, performance characteristics of the microprocessors used in the featured computers. This information is obtained from the IC manufacturer's catalog.

Component manufacturers produce a large number of products. Their paper catalogs are substantial. These catalogs must be converted into structured databases in which the attributes of every single product are stored in individual fields. This can be an overwhelming task. Fortunately, there are several information aggregators who have already created such databases using the printed catalogs of component manufacturers. Although these databases are not current (because they were created from the paper catalogs), they can be a good starting point for manufacturers. Several component manufacturers are currently negotiating the purchase of their data from information aggregators. Once they create their own databases, manufacturers will update the information themselves. As an example, the following table shows a list of connectors and two of their attributes. Typically, products have around ten such attributes. This is the core information in a catalog. Other information includes free text description, features/benefits, and product pictures.

Part Number	**Number of Contacts**	**Gender**
1298	5	Male
2145	9	Female
3410	15	Male
4156	15	Female

TABLE 1. Sample attributes of electronic connectors

Systems Manufacturers

Systems manufacturers face a different problem. Although they do not have a large number of products like component manufacturers, most of the products they offer are configurable. Configuration rules are used to express compatibility between different components of the system. Configuration rules are more complex to store and to manage than product attributes. They are also more difficult to communicate. Configurator software vendors use proprietary data structures to store configuration rules. Currently, there are no known efforts to standardize the definition of these configuration rules in order to communicate them through the supply chain. Instead, systems distributors refer their customers to the configurators on the manufacturers web sites. Systems manufacturers can only communicate product specifications of the individual products they sell, not the configuration rules of complete systems. Examples of individual products include printers, monitors, network cards, scanners. The following diagram shows that simple products, such as electronic components, need catalog functionality where a large amount of data can be stored, while complex products, such as routers and computers, need configurator functionality where less data but more rules are stored.

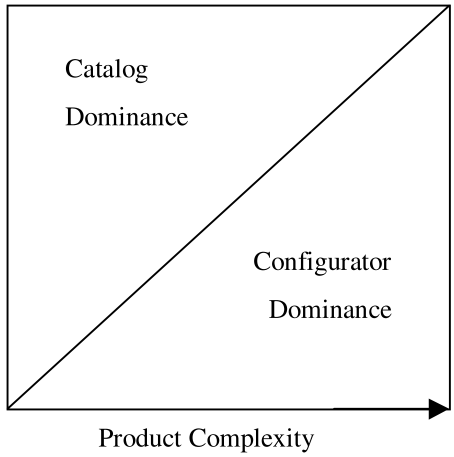

FIGURE 2. Simple products require catalog functionality while complex products require configurator functionality.

Sending Information to Trading Partners

Manufacturers need to send product information to their distributors and customers. One major issue that complicates this process is that different manufacturers use different terminology to describe their products. For example, one connector manufacturer will refer to the number of pins in a connector as “Number of Contacts”, while another connector manufacturer will refer to the same attribute as “Number of Positions”. A computer manufacturer will refer to the speed of a notebook computer as “Clock Speed” while another manufacturer will refer to it as “CPU Speed”. Distributors and customers who aggregate information from different manufacturers need to map product attributes into their own vocabulary. Several terminology standardization efforts are currently under way in different industry sectors. The RosettaNet project, for example (www.rosettanet.org), aims at standardizing information exchange in the information technology industry.

Another issue in information exchange is the format of the exchanged information. XML is certainly gaining momentum as a de-facto standard format for information exchange. The following diagram shows a manufacturer catalog communicating with a user's browser in HTML, and communicating with a distributor's catalog in XML.

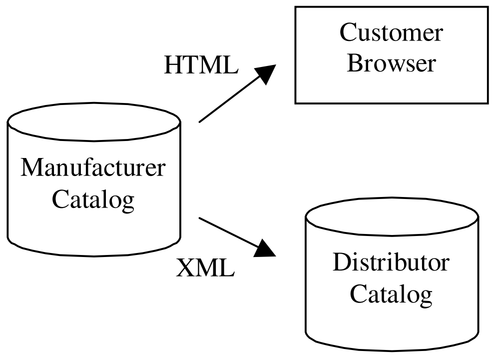

FIGURE 3. Electronic catalogs must be able to respond to a user request in HTML and to an agent request in XML

Designing a Generic Catalog Data Model

Typical product information databases found in ERP (Enterprise Resource Planning) systems contain the product number, a short description, and price. The product description field in these databases is meant to appear on an order or an invoice in order to associate a brief description to the product number, not to provide a full description of the product. In a catalog database, product attributes are individually stored. This is essential to perform parametric search (search by attribute value). Storing product attributes in addition to the brief description creates a new level of complexity because different product families have different attributes. It would be too difficult to manage multiple database tables with different column headers for different product families. Therefore, the catalog database must be designed to accommodate any product type without the need to create or modify tables. This can be accomplished by storing attributes in the form of name/value pairs instead of using one table column per attribute.

The following table shows product attributes where each attribute is a column header:

Part Number	**Number of Contacts**	**Gender**
1298	5	Male
2145	9	Female
3410	15	Male
4156	15	Female

TABLE 2. Database table contains one column per attribute.

The following table shows the same product attributes organized in name value pairs:

Part Number	**Attribute Name**	**Attribute Value**
1298	Number of Contacts	5
1298	Gender	Male
2145	Number of Contacts	9
2145	Gender	Female
3410	Number of Contacts	15
3410	Gender	Male
4156	Number of Contacts	15
4156	Gender	Female

TABLE 3. Database table contains attributes in the form of name/value pairs. This form allows storage of attributes for any product type in the same table.

The Issue of Multiple Languages

Global enterprises publish their product information in multiple languages. The translation effort can be significant. When information is managed in a database, the translation effort can be dramatically reduced. This is accomplished by creating a master list of product attributes and assigning a unique code to each attribute. Each code is then translated once into each supported language. When a new product is added, it is assigned a set of attributes from the master list. The new product data sheet can then be displayed in any of the supported languages by drawing from the master list of attributes without having to manually create attribute translation for this particular product.

The following table shows product attributes shown in coded name/value pairs:

Part Number	Attribute Name Code	Attribute Value Code
1298	1	5
1298	2	1
2145	1	9
2145	2	2
3410	1	15
3410	2	1
4156	1	15
4156	2	2

TABLE 4. Product attributes are stored in coded name/value pairs

The following table shows the feature codes translated into different languages

Attribute Name Code	Language	Value
1	English	Number of Contacts
1	French	Nombre de Contacts
2	English	Gender
2	French	Genre

TABLE 5. Attribute name codes are then represented in different languages in a separate table. Attribute value codes are translated in a similar table.

Catalog Integration with Other applications

A major requirement of catalog management software tools is their ability to integrate with other applications. For example, sales tax calculation can be quite complex and might necessitate the use of a specialized application. The catalog management tool should therefore provide means to invoke external applications.

Also, the catalog management tool should be able to communicate with back end applications (such as ERP) in order to access data such as product availability.

Architecture of an Online Catalog

Because of the need to integrate different applications in e-commerce sites, component-based software is a recommended architecture for the catalog management software. This architecture allows easy integration of best-of-breed components from multiple vendors into a complete e-commerce solution. Microsoft's COM/ASP (Component Object Model / Active Server Pages) platform, for example, provides a powerful environment for the creation of component-based e-commerce applications. The following diagram shows a catalog application integrated with a tax module and a connector component to the ERP system using the Microsoft COM/ASP architecture. The Active Server Pages shown here provide a scripting language that can be used to integrate the different components.

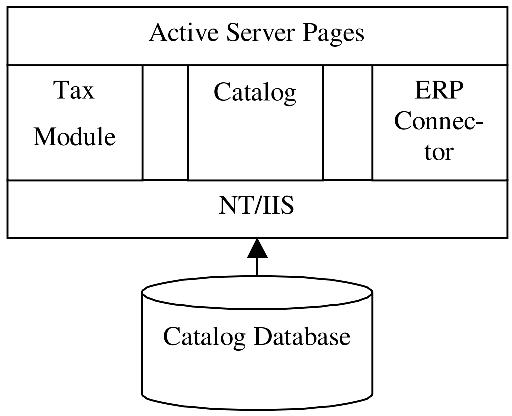

FIGURE 4. In a Microsoft COM/ASP environment, the catalog, tax module, and ERP connector are all COM objects. Scripts in the Active Server Pages are used to integrate the different COM objects and interface with the user.

Customizing the Page Layout of the Site

Another need in online catalog is the ability to customize the presentation of the web pages in order to maintain the company's identity. This is typically accomplished with HTML templates.

e-commerce vendors usually provide their own proprietary template functionality. However, in Microsoft's COM platform, Active Server Pages are used to create the HTML templates. The advantages of Active Server Page technology are that 1) it provides a rich scripting language to control the HTML page layout, and, 2) it can be used to "assemble" COM components from different vendors without the need to learn the individual template language of each vendor.

Catalog Authoring

Creating a catalog database starts by grouping products into families. Examples of computer product families are printers, scanners, monitors, servers, etc. Once product families are created, the list of attributes for each family is determined. For example, the attributes for printers could be:

Print Technology: Laser, Inkjet

Print Speed: 6PPM, 8PPM, 12PPM

Paper Size: Letter, A4, 11x17

After the attributes and their values are identified, the different attributes of each product in a family can be entered in a spreadsheet (similar to Table 2). Each family is entered in a separate spreadsheet and each product in the family is entered in a separate row. The spreadsheet can then be imported into the catalog database in a format similar to Tables 4 and 5.

Once the bulk of the data is entered, updates should be performed using a browser. If some of the catalog information already exists in other systems such as the ERP database, this information can be regularly imported using a custom import software. The information can then be complemented using the spreadsheet technique or a browser.

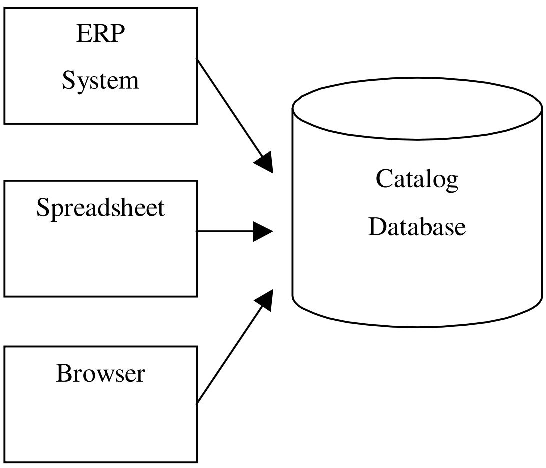

FIGURE 4. Product information can be imported for the ERP system and complemented using a spreadsheet import or a browser.

Product Search Techniques

After entering in an electronic catalog, the first step in product search is to identify the desired product family. This can be accomplished with a hierarchical menu, a picture menu, or with a keyword search. Once the family is identified, parametric search can help identify the right product within that family. Parametric search consists of displaying the list of attributes of the product and allowing the user to select the desired attribute values. The search returns either the set of matching products or a revised set of selectable attributes based on previous selections. If a revised set of attributes is returned, the user can narrow the selection further by selecting more attribute values and re-issuing the search. This step by step process known as Step Search ® can continue until a single product is found. During the parametric search process, the user should be able to display side by side comparison tables. Once a product is identified in a comparison table, the user should be able to display its detailed data sheet, and add it to a shopping cart. Note that all this information is displayed in HTML pages dynamically generated from the database. Also note that if the user knows the product part number when entering the catalog, she should of course be able to display directly the data sheet of that product.

On-the-Fly Catalog Aggregation

With the evolution of agent technology, and with the continuous increase in the Internet bandwidth, information aggregation will be performed on-the-fly. In other words, instead of building and maintaining their own catalogs, distributors, for example, will let their customers select product attributes on an HTML form, and will then send agents to perform queries directly on the manufacturers databases. The information collected from different manufacturer databases is then aggregated on the fly and presented to the user. The advantage of this technique is that distributors do not have to maintain a database, and they are guaranteed that the information collected from the manufacturers sites is up-to-date. The disadvantage of this solution is that the process can take a long time to execute.

Conclusion

The move from static publishing to dynamic, database-driven publishing of product information is inevitable in a world where supply chain integration is happening at a very fast pace.

The need to exchange product information at a high pace will force direct communication between the servers of trading partners without human intervention.

Unfortunately, some businesses such as distributors, will have to wait for manufacturers to create their product information databases before they can create their own. They also have to wait for terminology standards and communication standards to be established before they can automatically aggregate multi-vendor product information.

Once the problem of data structuring is solved and electronic catalog use will be widespread, the process of free catalog navigation and adding items to a shopping cart will be complemented with a "Sales Process Automation" approach where buyers will be guided through a wizard to build a complete offer. This process will more closely resemble the interaction with a sales person.

References

Books:

Building Database-Driven Web Catalogs Sherif Danish, Patrick Gannon, McGraw Hill. 1998

Electronic Commerce : A Manager's Guide - Ravi Kalakota, Andrew B. Whinston (Contributor); Paperback

Business-To-Business Internet Marketing : Proven Strategies for Increasing Profits Through Internet Direct Marketing Barry Silverstein / Paperback / Published 1998

Web Catalog Cookbook, Cliff Allen, Deborah Kania

Professional Active Server Pages 2.0 - Brian Francis, et al; Mass Market Paperback

Web Sites:

Saqqara Systems, Inc. www.saqqara.com

IBM www.ibm.com (search for Net.Commerce)

Open Market (www.openmarket.com)

Broadvision www.broadvision.com

ICat (www.icat.com)

Interworld (www.interworld.com

Intershop www.intershop.com

Veo Systems www.veo.com