# Modeling Autonomous Catalog for Electronic Commerce

Yuan-Chi Chang, Vamsavardhana R. Chillakuru, and Min Wang

IBM Thomas J. Watson Research Center,
P.O. Box 704, Yorktown Heights, NY 10598, USA

The catalog function is an essential feature in B2C and B2B e-commerce. While catalog is primarily for end users to navigate and search for interested products, other e-commerce functions such as merchandising, order, inventory and aftermarket constantly refer to information stored in the catalog [1]. The billion-dollar mail order business was created around catalog long before e-commerce. More opportunities surface after catalog content previously created on paper is digitized. While catalog is recognized as a necessity for a successful web store, its content structure varies greatly across industries and also within each industry. Product categories, attributes, measurements, languages, and currency all contribute to the wide variations, which create a difficult dilemma for catalog designers.

We have recently encountered a real business scenario that challenges traditional approaches of modeling and building e-catalog. We were commissioned to build an in-store shopping solution for branches in retail store chains. The local catalog at a branch is a synchronized copy of selected enterprise catalog content plus branch specific information, such as item location on the shelf. A key business requirement, which drives up the technical challenge, is that the in-store catalog solution needs to interoperate with the retail chain's legacy enterprise catalog or its catalog software vendor of choice. This requirement reflects the business reality that decisions to pick enterprise software and branch software are usually not made simultaneously nor coordinated. As we learned that hundreds of enterprise catalog software, legacy and recent, is being used in industries such as grocery, clothing, books, office staples and home improvement, our challenge is to create a catalog model that is autonomously adapting to the content of enterprise catalog in any of the industries.

A straightforward answer to the challenge is to build a mapping tool that will convert enterprise catalog content to the pre-designed in-store catalog, but this approach is highly undesirable. The difficulty lies within that it is impossible to predict the content to be stored. A simple example to illustrate the difficulty is by looking at what is stored in catalog for Home Depot, a home furnishing retailer, and by examining what is stored in catalog for Staples, an office equipment retailer. A kitchen faucet sold at Home Depot has information about its size, weight, material, color, and style. On the other hand, a fax machine sold at Staples carries attributes such as speed, resolution, and tone dialing. These attributes need to be stored in the catalog for retrieval and product comparisons. Without knowing where a catalog will be used, our design obviously

P. Atzeni et al. (Eds.): ER 2004, LNCS 3288, pp. 828–830, 2004.

cannot pre-set the storage schema for either faucets or fax machines. Needless to say, there are hundreds of thousands of products whose information needs to be stored in catalogs. Today's catalog solutions in the market also suffer from over design, which leads to wasted storage space, over-normalized schema and poor performance. Multi-language support, currency locale, geological label and access control are commonly embedded and inseparable from the main catalog functions. Suppose a company only operates stores in California. The additional features can turn highlights to burden.

Further enhancing the shortfall of the traditional catalog modeling and mapping approach is the lack of configurability and optimization. Customization made on small delta changes to the catalog data model propagates in a magnified way all the way up to business logic and presentation layers. Furthermore, the vertical schema to store catalog attributes in name-value pairs distorts database statistics and makes catalog queries hard to optimize [3] [4]. We foresee no easy way to continue the traditional methodology for a satisfactory solution to our problem.

In this paper, we propose a set of abstracted catalog semantics to model an autonomous catalog to become the in-store catalog solution. An autonomous catalog exhibits two key properties of autonomic computing: self-configuration and self-optimization [2]. It receives definitions of catalog entities from enterprise catalog to synthesize and create persistent storage schema and programming access interface. It buffers objects for cached retrieval and learns from search history to create index for performance. The use of autonomous catalog requires little learning and training since it morphs into enterprise catalog content structure. Changes can be reflected instantly at storage schema and programmatic interfaces.

We model this autonomous catalog by associations of basic categorical entities. A categorical entity is defined as a named grouping of products that share similar attributes. Instances of a categorical entity are physical, procurable products or services. For example, the kitchen faucet may be declared as a categorical entity and one of its instances is Moen Asceri. A categorical entity may be pointing to one or more categorical entities to establish parent or child category relationship. Attributes in a categorical entity may be completely different from those in another and yet in both cases, they are efficiently stored in a normalized schema without applying the vertical schema.

We define five operations including add, update, delete, search and retrieve on categorical entity. To shield software developers from accessing instances of categorical entities directly, these five catalog operations can only be executed through a programming language interface such as Java. When a new entity is declared by the enterprise catalog in XML Schema expression, new Java classes and interfaces, following a predefined template of these five operations, will be automatically synthesized.

For example, the enterprise may declare an entity named 'Kitchen Faucet' with five attributes. Our autonomous catalog then creates tables in the database to store instances of faucets and synthesizes a Java class with methods to populate, retrieve and search the instances by attribute values. Kitchen faucet may be associated with plumbing and kitchen categories. The Java class has methods to support searches from the associated categories. Revisiting the aforementioned catalog features such as multi-language support, we can easily add new attributes describing kitchen faucet in foreign languages applicable to use cases. There is no unused space for catalog attributes not needed.

Another advantage of the autonomous catalog is its ability to capture more sophisticated modeling semantics at runtime, due to the flexibility of programming language wrapper. For example, in the synthesized Java class, programmatic pointers can reference an external taxonomy or ontology for runtime inferencing. Catalog content linked to a knowledge management system can support more intelligent queries such as 'which kitchen faucets are recommended for water conservation?' This further brings catalog modeling beyond the inclusive entity-relationship diagram.

The modeling of autonomous catalog enables it to re-configure itself while administrators and programmers are shielded from knowing the details in managing the flexible persistent storage. As the Java classes change and evolve to adapt to the enterprise catalog content, one can envision that business logic that invokes these Java classes to be modeled and generated autonomously as well. We are investigating the modeling of merchandising and order tracking to demonstrate the feasibility of autonomous modeling of business logic.

## References

1. S. Danish, "Building database-driven electronic catalogs," *ACM SIGMOD Record*, Vol. 27, No. 4, December 1998.
2. J. O. Kephart and D. M. Chess, "The vision of autonomic computing," *IEEE Computer Magazine*, Janurary 2003.
3. S. G. Lee, et.al. "An experimental evaluation of dynamic electronic catalog models in relational database systems," *2002 Information Resources Management Association International Conference*, Vol. 1, May 2002.
4. M. Wang, Y. C. Chang and S. Padmanabhan, "Supporting efficient parametric search of e-commerce data: a loosely-coupled solution," *8th International Conference on Extending Database Technology*, March 2002.