A Product Information System based on an Object-Oriented Internet Database System

A. Kobler
Institute for Information Systems
ETH
CH-8092 Zurich, Switzerland
kobler@inf.ethz.ch

M. C. Norrie
Institute for Information Systems
ETH
CH-8092 Zurich, Switzerland
norrie@inf.ethz.ch

Abstract

We present a product information system which supports enterprise-wide access to product structure data through the World-Wide Web (WWW). This information system provides ready access to data on products, their structures, variants and configurations. To meet the needs of various enterprises and types of user, the system is based on a product data model which is both flexible and extensible. The system has been implemented using the OMS object-oriented database management system and its generic WWW interface.

1. Introduction

Fundamental to the operation of global and virtual manufacturing enterprises is ready access to product data by a variety of employee types such as project managers, designers and salesmen. Such information includes the types of products available, product characteristics and details of product structures including variants and configurations.

We use the term *product information system* to refer to a shared repository of such product data thereby distinguishing it from *product data management systems* (PDMs) which have a more specialised, and more complex, role in supporting the manufacturing process itself. As we shall see, the product data model of a PDM is therefore typically less general than that of our proposed product information system in that it tends to be tailored to a particular type of user and activity.

A product information system is not a replacement for a PDM, but rather serves a different and complementary purpose in supporting the general product information requirements of the various departments of an enterprise. Further, it may also serve as the central product information broker across enterprises providing information about the products of various enterprises and links to product catalogues of individual enterprises.

The World-Wide Web (WWW), together with its browsers such as Netscape Navigator, provides a convenient means of accessing a product information system both within and across enterprises via Intranet and Internet technologies. It is therefore appropriate that such a product information service be based on a database system with a WWW interface.

In this paper, we describe a product information system that has been developed using an object-oriented database management system, OMS [6]. The underlying product data model is both flexible and extensible and combines advantages of various existing PDM product data models. Internet (and Intranet) access is provided through Internet OMS which is a generic WWW interface to OMS supporting both object browsing and ad-hoc database queries.

In Section 2, we describe the main requirements of an enterprise-wide product information system. Section 3 then details the product data model we have developed. The implementation of our product information system based on the Internet OMS object-oriented database management system is described in Section 4. Section 5 outlines future work and concluding remarks are given in Section 6.

2. Requirements

The various users within an enterprise have different product information requirements. A designer may wish to identify products with particular characteristics that can be used in his design. A marketing manager may require information on product concepts before the products have even been designed and therefore the structure is unspecified. A salesman wishes to be able to configure a particular product for his customer by selecting the appropriate variants, parameters and options. Further, these various users may have a wide geographical distribution and drastically different computing platforms. In the case of a salesman, he may use a mobile computer and require access to product information via the Internet.

Clearly, these problems of user, information and platform heterogeneity are even greater in the case of virtual or global enterprises. It is increasingly the case that information services are being placed on the Internet via the use of WWW interfaces to database systems, thereby providing a platform-independent means of remote access to data. Within an enterprise, the benefits afforded by WWW may be provided on an Intranet basis. A global or virtual enterprise is likely to require Internet access. In the context of this paper, we use the term Internet to cover both cases since we focus only on WWW technologies which are common.

To meet the requirements of the various users, we propose an Internet database system for general product information. An Internet database system is one which has a WWW interface that provides access via the Internet through one of the usual WWW browsers, such as Netscape Navigator. The product information system we have developed uses our own object-oriented database system OMS [6, 4]. The OMS system has a number of advantages with respect to the development of such an Internet product information system. First, the underlying object-oriented data model OM provides rich modelling constructs in terms of structures and constraints which lead to a simple, yet expressive product data model. Second, the use of an object-oriented model provides a convenient way to form an integration database with links to other application systems via methods. The OMS system also supports a full query language. Further, Internet information services are supported through a generic WWW interface to OMS. This contrasts with many existing WWW interfaces to database management systems which require an application specific interface to be developed using queries embedded in HTML pages. The Internet OMS interface mimics directly the general OMS browser. Clearly, however, it can also be adapted to produce an application specific interface.

The underlying product data model has to be general enough to support the information needs of the various user groups, and flexible enough that it can be adapted to the requirements of a particular enterprise. We developed our own product data model which allows both the concept of a product, as well as details of product structures and configurations, to be represented. Further, the set of product characteristics are not fixed, but rather are modelled explicitly by the user allowing new characteristics and their associated value sets to be added and updated as required. This contrasts with the product data models of existing PDM systems which tend to be less flexible. For example, [2] allows a product to be represented in the database independently of its structure, but assumes a fixed set of product characteristics which can only be modified by using a special tool for changing the database schema.

Additionally, a product information system should support not only the representation of product variant structures and configurations, but also the configuration process itself.

In the following section, we present our product data model in detail before continuing in later sections with a description of our system.

3. Product modelling

An essential and crucial aspect of any product information system is its underlying product data model. This model should support not only the description of various product structures, but also the process of product configuration. Hence, it follows that such a product data model should have characteristics such as

- *Flexibility*
  It should not be a problem to model different kinds of complex structured products and their variants.
- *Extensibility*
  No major redesign of the model should be necessary if new requirements have to be fulfilled.
- *Portability*
  The integration of the data model into specific application domains should be possible.

Another important issue in this context is the selection of an appropriate data modelling concept. Our proposal for a simple, but powerful, product data model is based on the generic object data model OM. A short description of OM is given in the next section.

In our product data model, we distinguish and support four main activities which are necessary to provide user modelling of a product on different conceptual levels:

- *Product Classification*
  A convenient way of expressing the more conceptual view of products is by classifying them according to certain criteria into different product groups, respectively, product families.
- *Product Structuring*
  The result of this activity is usually a hierarchy of components. This hierarchy is described best with some sort of *part_of* relationship between products.
- *Product Characterisation*
  The characteristics of a product such as *size*, *length* and *colour* have to be identified together with their values.

- *Product Description*
  Product Descriptions such as *Rules*, *Constraints*, *Formulas* and *Queries* are required for specification of the overall product structure, and for supporting the product configuration process.

3.1. Overview of the OM model

In this section we want to give a brief description of the generic object data model *OM* which specifies constructs and operations for the management of interrelated collections of objects. The *OM model* is generic in that it is independent of the form of individual objects and hence of the underlying type model and programming language system.

Data modelling can be seen as the construction of conceptual models of reality. It is therefore involved with the description of general concepts of the application domain. On the other hand, a database schema is considered not only as a description of reality but also as a description of the representation of that reality in the database.

The *OM model* distinguishes these two forms of description through the concepts of classification and typing. The former corresponds to classifying entities into categories, the latter corresponds to describing the representation of entities within the database in terms of types (figure 1).

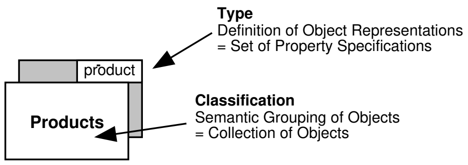

Figure 1. Classification and types

The *OM model* specifies various collection constructs and structural constraints. Specific support is given for the representation of classification structures and for binary associations of collections (Figure 2 and 3). Further, the model specifies a general set of operations over collections which we refer to as a collection algebra.

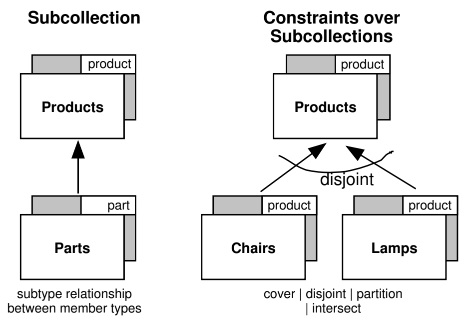

Figure 2. Classification structures

A more detailed description of the *OM model* can be found in [6, 5, 4].

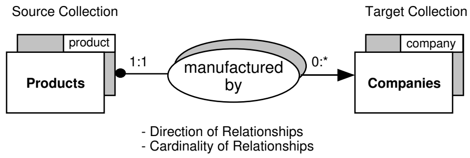

Figure 3. Associations

3.2. Product classification and product structure

The part of the product data model shown in figure 4, addresses the modelling and classification of complex structured products as well as of product variants.

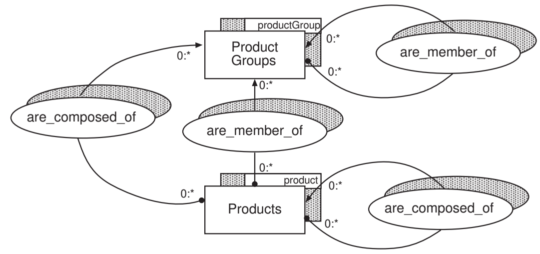

Figure 4. Product classification and structure

The collection `Products` contains all products which are either *composed products* or *single part products*, whereas the objects in the collection `Product Groups` represent *product groups*, respectively, *product families*.

The **classification** of products can be managed by the association `are_member_of` between the collection `Products` and the collection `Product Groups`. Since there is also an association `are_member_of` from the collection `Product Groups` to itself, it is possible to have a rather complex classification hierarchy.

Consider, for example, a company producing and selling three different models of chairs: One for the garden, one for the living room, and one for the kitchen. Each model can be ordered in various variants.

To classify these products, we can first create three *product group* objects “Garden Chair”, “Living Room Chair”, and “Kitchen Chair” which represent the models of chairs. These are then put into the collection `Product Groups`. Now we create *composed product* objects for each product variant, and put these into the collection `Products`. Finally, we make associations between the *composed products* and their corresponding *product group*.

The **structure** of products can be modelled by the two associations `are_composed_of`. The one from the collection `Products` to itself expresses the fact that products can be part of other products, respectively, are composed of other products. The one from the collection `Products` to the collection `Product Groups` further extends the flexibility of structuring products, especially product variants.

3.3. Product characteristics

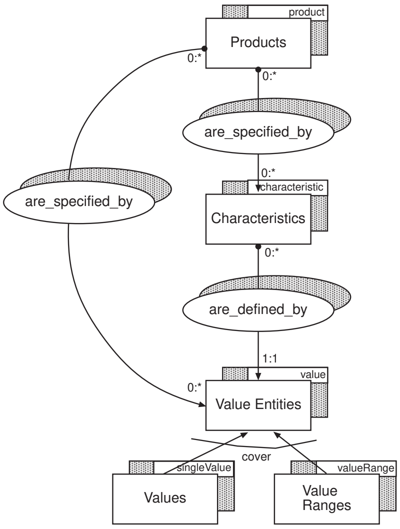

Figure 5. Product characteristics

All products can be specified by one or more characteristics such as *size*, *colour* and *price*.

In our product data model, all *important* characteristics are defined as independent entities and not as attributes of a product object. Therefore, we are very flexible in modelling different kinds of products and it is possible to add and remove characteristics without having to change the type specification of the product objects. An *important* characteristic can be regarded as a characteristic which is either common for many products such as *colour* or which is significant for characterising a specific product.

One of the associations `are_specified_by` relates characteristics to products. Each characteristic is defined by one or more value entities. This is managed by the association `are_defined_by`. As in the case of characteristics, we represent *important* values as entities and not as attributes in our data model. We are convinced that this approach is not only appropriate for modelling product variants, but also is an advantage for the product configuration process.

A value entity has a type (e.g. integer, string) and a unit of measurement (e.g. cm, kg). A value entity is either a single value or represents a range of values (e.g. 1.5 - 2.5). Not all characteristics need to be specified by a value entity. A characteristic can also be evaluated by a formula (as described later). The other association `are_specified_by` assigns value entities to products.

3.4. Product Description

For describing the overall structure of a product, as well as for supporting the process of product configuration, we need a mechanism which allows the formulation and evaluation of facts such as "the minimum and maximum number of each component", "component is mandatory, respectively optional", "characteristic *volume* is calculated by a formula", "all parts must have the same colour", and so on.

We introduce therefore the concept of *Product Descriptions*. A product description can either be a rule, a constraint, a formula or a query (figure 6).

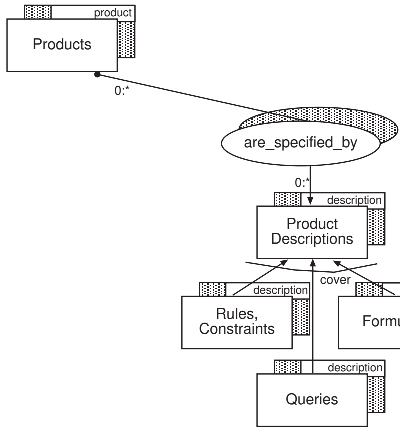

Figure 6. Product descriptions

All product descriptions can be defined using the query language of our database system OMS together with the *Product Data Description Language* (PDDL) which we are currently developing. This makes it possible to create very small yet expressive product descriptions, and since products are related to these descriptions by the association `are_specified_by`, every product "knows" exactly how to configure itself.

The *Product Descriptions*, respectively their *PDDL* representations, are interpreted and evaluated at run-time, for example during the configuration process of a product. This contrasts, for example, with the language *EXPRESS* which is part of the ISO 10303 standard *STEP* (International Standard for the computer-interpretable representation and exchange of product data) and which is a language for specifying product data in a more static way and is not meant to be an entire programming language. Constructs like "if..then..else", "while...do", and the use of variables are also provided by *PDDL*. Since we did not want to define a whole new programming language, we have taken *Oberon-0* [10] as a basis and extended it with the constructs needed for describing product data.

4. Internet OMS

A first prototype of our product information system for verifying our product data model has been implemented using the object-oriented database system OMS [6] which has been developed in our research group. OMS can be considered as a system supporting an instantiation of the OM generic data model presented in section 3.1. An Internet interface to OMS, together with an agent-based client/server-architecture consisting of several Java applets, has also been developed [3], thereby making it possible to access the database using a Java capable WWW browser.

The following example illustrates query interaction between a user and the system. It refers to our product data model as well as to the example described in section 3.2.

Suppose, we want to find all the kitchen chairs available in the colour blue. First we have to formulate a query using the OMS query language *AQL*:

```
// selection of all products which are member
// of the product group 'Kitchen Chair'
(domain (
  product_member_of_group
    range_restriction
      select group in ProductGroups
      having (group@name) = "Kitchen Chair"))
// intersection with the...
intersect
// selection of all products that are related
// to the value object 'blue'
(domain (
  product_spec_by_valueEntity
    range_restriction
      select colour in ValueEntities
      having (colour@value) = "blue"))
```

This query can be entered interactively in a special query window either of the WWW browser or of the OMS system. The system then presents the matching products as a list in a new window. Note that all windows showing information about product data are created dynamically by the system. Double-clicking on one of the matching products in the list opens a new window showing the selected product object (figure 7).

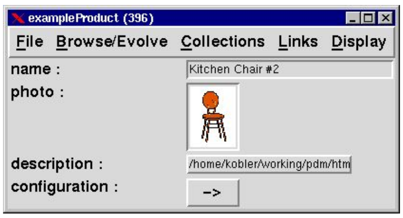

Figure 7. A product object

A product object, as shown in figure 7, can also have properties which represents links, for example to HTML pages, or which are methods.

Of course, it is not the intention that a user has to type in such complex queries when he accesses the system. Instead, the product information system provides functionalities based on pre-defined queries and also to assist the user in ad-hoc querying. Since OMS is an object-oriented database, objects can provide methods which also are executable remotely over a network connection. It is further possible to include queries within methods. These features of OMS, among many others, make it possible to build small but very powerful Internet applications with minimal application programming effort.

5. Future work

Our plans for the future especially focus on the following three topics:

- *Extension of the Product Data Model*
  The definition of our *Product Data Definition Language* PDDL has yet to be completed. Another important issue for a product data model is the necessity of being able to express the temporal aspects of product data, for example for managing the history of products. Since there already exists a temporal extension to *OM* [8], the integration of temporal aspects should be relatively straightforward.
- *Product Configuration Process*
  The *Product Configuration Process*, as we see it, can be divided into three different tasks: The configuration
  - of *Single Part Products*,
  - of *Composed Products*, respectively, *Product Variants*, and
  - of *Product Groups*, respectively, *Product Families*

  We hence want to extend our prototype system to provide these tasks and to give the user support for the process of composing and configuring products.
- *OMS/Java*
  OMS is a powerful database system designed for rapid prototyping. Another instantiation of the generic object model *OM* is currently under development for the *Java environment*. The *OMS/Java* database management system will include features such as
  - Multi-user Capabilities
  - Client/Server Architecture
  - Internet Interface
  - Remote Method Invocation Mechanism
  - Built-in support for the evaluation of *Product Descriptions*

After having developed this system, we will port our prototype to *OMS/Java*. In this system, methods will be able to execute on the client side and not only on the server side as at present.

6. Conclusions

We have presented a product data model which is extensible and flexible enough to fulfil today's increasing complexity of product structures. We have also outlined the requirements for a modern product information system and have given an overview of our own prototype system.

In addition, our proposed product data model addresses the fact that it will become more and more important to have a general model which is suitable for different kinds of user profiles as well as for various application domains. The integration of existing data sources together with the demand for wider access to data are also significant issues we have taken into account with our data model.

To compare our product data model to existing ones described in ([7, 1, 9]), we can emphasise the following key features:

- *Flexibility*
  With our approach, it is possible to describe complex product structures and expressive classification hierarchies with only a single model. This allows the modelling of product variants to be done on several conceptual levels.
- *Extensibility and Portability*
  In particular, the concept of *Product Descriptions* and the idea of modelling *Product Characteristics* as entities instead of attributes makes the model very adaptable for new requirements and also allows a smooth integration into various application domains.
- *Object-Oriented Data Model*
  One reason why object-oriented data models are becoming increasingly popular is the possibility to specify the behaviour of entities and hence to encapsulate functions within objects. Another reason lies in the extended modelling capabilities in terms of typing and classification. The generic object model *OM* chosen for representing our product data model makes a clear delimitation between the object representation (Typing) and the semantic grouping of objects (Classification), and can be regarded as a powerful tool for modelling complex data structures.
- *Different Levels of Modelling*
  The process of classifying, of structuring, and of characterizing a product can be seen as different levels of data modelling, or as different abstract views of the data. We believe that it is really necessary to make this kind of abstraction for the reason of simplifying product data representation. We are convinced that our proposed product data model makes it much simpler to manage product data.

To date, our prototype system can be characterized as a system that provides

- Internet access through a generic Internet interface,
- access to the database by using a standard WWW browser,
- capabilities for rapid prototyping of object-oriented Internet applications,
- a powerful query language, and
- support for modelling complex structured product data.

By extending our data model with the notion of *Product Descriptions* and by porting the prototype system to the Java environment, our product information system will become more platform independent and even more open to all sorts of application demands.

References

[1] B.-C. Björk. *Requirements and information structures for building product data models*. PhD thesis, Helsinki University of Technology, Finland, Dec. 1995.

[2] Eigner+Partner GmbH, Ruschgraben 133, D-76139 Karlsruhe, Germany. *CADIM/EDB*.

[3] A. Erni and M. C. Norrie. SnowNet: An Agent-Based Internet Tourist Information Service. In *Proc. of the 4th Int. Conf. on Information and Communications Technology in Tourism (ENTER'97)*, Edinburgh, Scotland, Jan. 1997.

[4] M. C. Norrie. An Extended Entity-Relationship Approach to Data Management in Object-Oriented Systems. In *12th Intl. Conf. on Entity-Relationship Approach*, pages 390–401, Dallas, Texas, Dec. 1993. Springer-Verlag, LNCS 823.

[5] M. C. Norrie. Distinguishing Typing and Classification in Object Data Models. In *Information Modelling and Knowledge Bases*, volume VI, chapter 25. IOS, 1995. (originally appeared in Proc. European-Japanese Seminar on Information and Knowledge Modelling, Stockholm, Sweden, June 1994).

[6] M. C. Norrie and A. Würgler. OMS Object-Oriented Data Management System. Technical report, Institute for Information Systems, ETH Zurich, CH-8092 Zurich, Switzerland, 1997.

[7] S. Schwarze. *Configuration of Multiple-Variant Products: Application Orientation and Vagueness in Customer Requirements*. PhD thesis, Swiss Federal Institute of Technology, Zurich, Aug. 1996.

[8] A. Steiner and M. C. Norrie. Temporal Object Role Modelling. In *Conference on Advanced Information Systems Engineering (CAiSE)*. Barcelona, July 1997. to appear.

[9] U. R. . B. N. . A. Storr. *Computer Integrated Manufactoring and Engineering*. Addison-Wesley, 1993.

[10] N. Wirth. *Compiler Construction*. Addison-Wesley, 1 edition, 1996.