Handling Conceptual Multidimensional Models using XML through DTDs*

Enrique Medina, Sergio Luján-Mora, and Juan Trujillo

Departamento de Lenguajes y Sistemas Informáticos
Universidad de Alicante. SPAIN
{emedina,slujan,jtrujillo}@dlsi.ua.es

Abstract. In the last years, several approaches have been proposed to easily capture main multidimensional (MD) properties at the conceptual level. In this paper, we present how to handle MD models by using the eXtensible Markup Language (XML) through a Document Type Definition (DTD) and the eXtensible Stylesheet Language Transformations (XSLT). To accomplish this objective, we start by providing a DTD which allows to directly generate valid XML documents that represents MD properties at the conceptual level. Afterwards, we provide XSLT stylesheets to automatically generate HTML pages that correspond to different presentations of the same MD model.

1 Introduction

Data warehouses, OLAP applications and MD databases, based on the MD modeling, provide companies with huge historical information for the decision making process. In the last years, there have been some proposals to accomplish the conceptual MD modeling of these systems [1][2][3]. Due to space constraints, we refer the reader to [4] for a detailed comparison and discussion.

In this paper, we present how to handle the representation, manipulation and presentation of MD models accomplished with the object-oriented (OO) approach presented in [5] by using a standard format. To accomplish this goal, we define a DTD to validate XML documents that store all the MD properties. Furthermore, we combine XSLT stylesheets and XML documents in a transformation process to obtain HTML pages.

The remainder of this paper is structured as follows: Section 2 describes the basis of the OO conceptual MD modeling approach that we adopt in this paper. Section 3 presents how to represent MD models with the XML and the XSLT. Finally, in Section 4 we present our conclusions and future works.

2 Object-Oriented Multidimensional Modeling

In this section, we summarize how an OO MD model, based on the Unified Modeling Language (UML), can represent main structural MD properties at the

* This paper has been partially supported by the Spanish Ministry of Science and Technology, Project Number TIC2001-3530-C02-02.

conceptual level. In this approach, the main structural properties are specified by means of a UML class diagram in which the information is clearly separated into facts (*fact classes*) and dimensions (*dimension classes*). Then, fact classes are specified as composite classes in shared aggregation relationships of n dimension classes. *Many-to-many* relationships between facts and particular dimensions are indicated by the *1..** cardinality on the dimension class role (Fig. 1 (a)). For nonadditive measures, additive rules are defined as constraints and are included in the fact class. Furthermore, derived measures can also be explicitly considered (indicated by /) and their derivation rules are placed between braces near the fact class (Fig. 1 (a)). This OO approach also allows to define identifying attributes in the fact class, by placing the constraint *{OID}* next to an attribute name. In this way we can represent *degenerate dimensions* [6] (Fig. 1 (a)).

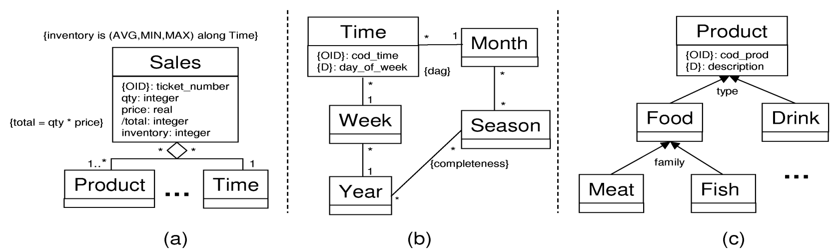

Fig. 1. Multidimensional modeling using the UML

With respect to dimensions, every *classification hierarchy* level is specified by a *base class*. An association of classes specifies the relationships between two levels of a classification hierarchy. The only prerequisite is that these classes must define a Directed Acyclic Graph (DAG) rooted in the dimension class (constraint *{dag}*). The DAG structure can represent both alternative path and multiple classification hierarchies. Every classification hierarchy level must have an *identifying* attribute (constraint *{OID}*) and a *descriptor* attribute (constraint *{D}*). The multiplicity *1* and *1..** defined in the target associated class role addresses the concepts of *strictness* and *non-strictness*. Moreover, defining the *{completeness}* constraint in the target associated class role addresses the completeness of a classification hierarchy (Fig. 1 (b)). Finally, the *categorization of dimensions* is considered by means of generalization-specialization relationships (Fig. 1 (c)).

3 Representing Multidimensional Models with XML

A relevant feature of a model should be its capability to share information in an easy and standard form. The XML [7] is rapidly being adopted as a specific standard syntax for the interchange of semi-structured data.

We have used the XML to store the MD models accomplished by our OO approach. The correct structure of the XML documents has been defined by means of a DTD. As an example of the DTD, the following fragment contains elements to express dimensions and their classification hierarchies:

```
<!ELEMENT DIMCLASSES (DIMCLASS+)>
<!ELEMENT DIMCLASS (ASOCLEVELS?, CATLEVELS?)>
<!ELEMENT ASOCLEVELS (ASOCLEVEL+)>
<!ELEMENT ASOCLEVEL (DIMATTS?, RELATIONASOCS?, METHODS?)>
<!ELEMENT DIMATTS (DIMATT+)>
<!ELEMENT DIMATT EMPTY>
<!ATTLIST DIMATT id ID #REQUIRED
   name CDATA #REQUIRED      atomic %Boolean; "true"
   type CDATA #REQUIRED      description CDATA #IMPLIED
   initial CDATA #IMPLIED    derivationRule CDATA #IMPLIED
   OID %Boolean; "false"     D %Boolean; "false">
```

```
<!ELEMENT RELATIONASOCS (RELATIONASOC+)>
<!ELEMENT RELATIONASOC EMPTY>
<!ATTLIST RELATIONASOC child IDREF #REQUIRED
   name CDATA #IMPLIED          description CDATA #IMPLIED
   roleA %Multiplicity; "1"     roleB %Multiplicity; "M"
   completeness %Boolean; "false">
<!ATTLIST ASOCLEVEL     id ID #REQUIRED
   name CDATA #REQUIRED         description CDATA #IMPLIED
   showAtts %Boolean; "true"    showMethods %Boolean; "true"
   autoResize %Boolean; "true">
```

Notice that attributes within these elements in the DTD allow to express all the MD properties at the conceptual level. For example, non-strictness may be defined by assigning the same value `M` to both attributes `roleA` and `roleB` in the DTD element RELATIONASOC, and completeness is defined by means of the DTD attribute `completeness`. Moreover, identifying and descriptor attributes within dimensions are defined using the DTD attributes `OID` and `D` in DIMATT.

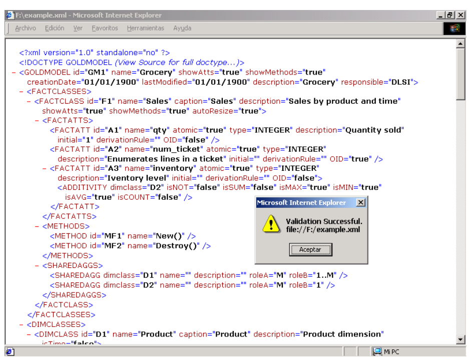

Fig. 2. An XML document displayed in Microsoft Internet Explorer

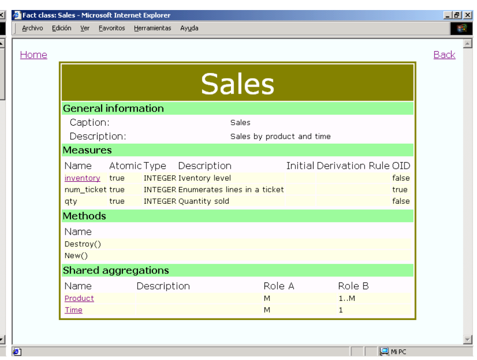

Fig. 3. Model navigation in the web using a browser

Fig. 2 shows the presentation of an XML document generated by our CASE tool in a popular web browser (*Microsoft Internet Explorer*). However, only some web browsers partly support the XML. As a consequence, we are currently forced to transform XML documents into HTML pages in order to publish them on the web. The best method to accomplish this task is the use of the XSLT [7], as a language for transforming XML documents. XSLT stylesheets describe a set of patterns (templates) to match both elements and attributes defined in a DTD, in order to apply specific transformations for each considered match.

We have used the XSLT processor *Instant Saxon 6.5*. This processor is full compliance with XSLT 1.0 and also allows the use of some features from XSLT 1.1, such as `xsl:document` (this new instruction allows to create different outputs from the same XML document). Our XSLT stylesheet generates a collection of HTML pages with links between them.

The resulting HTML pages allow to navigate through the different presentations of the model on a web browser. All the information about the MD properties of the model is represented in the HTML pages. For example, in Fig. 3 the first page is shown as an example of navigation for one of the presentations that contains the definition of the Sales fact class. At the bottom, we can notice the name of the dimensions: Product and Time. These are links that allow to navigate along the pages that contain the definition of those dimensions.

4 Conclusions and Future Works

We have presented how to store MD models accomplished by an OO conceptual approach, based on the UML, in XML documents. We have defined a DTD and we have provided XSLT stylesheets that allow to generate different presentations of the same MD model on an HTML format. In this way, we have provided a standard format to represent main MD properties at the conceptual level.

We are considering to adapt our proposal to the new emerging standards. In this sense, we are currently working on providing an XML Schema instead of a DTD. With respect to the presentation, XSL FO can be used to specify in deeper detail the pagination, layout, and styling information of XML documents. However, there are no current tools that completely support XSL FO.

Furthermore, we are currently studying the Common Warehouse Metamodel (CWM), a Meta-Object Facility (MOF) compliant metamodel, as a common framework to easily interchange warehouse metadata. CWM stores metadata in the XML Metadata Interchange (XMI) format. This proposal provides designers and tools with common definitions but lacks the complete set of information an existing tool would need to fully operate.

References

1. Golfarelli, M., Rizzi, S.: A methodological Framework for Data Warehouse Design. In: Proc. of the ACM 1st Intl. Workshop on Data warehousing and OLAP (DOLAP'98), Washington D.C., USA (1998) 3–9
2. Sapia, C., Blaschka, M., Höfling, G., Dinter, B.: Extending the E/R Model for the Multidimensional Paradigm. In: Proc. of the 1st Intl. Workshop on Data Warehouse and Data Mining (DWDM'98). Volume 1552 of LNCS. (1998) 105–116
3. Tryfona, N., Busborg, F., Christiansen, J.: starER: A Conceptual Model for Data Warehouse Design. In: Proc. of the ACM 2nd Intl. Workshop on Data warehousing and OLAP (DOLAP'99), Kansas City, Missouri, USA (1999)
4. Abelló, A., Samos, J., Saltor, F.: A Framework for the Classification and Description of Multidimensional Data Models. In: Proc. of the 12th Intl. Conference on Database and Expert Systems Applications (DEXA'01), Munich, Germany (2001) 668–677
5. Trujillo, J., Palomar, M., Gómez, J., Song, I.Y.: Designing Data Warehouses with OO Conceptual Models. IEEE Computer, special issue on Data Warehouses **34** (2001) 66–75
6. Kimball, R.: The data warehousing toolkit. 2 edn. John Wiley (1996)
7. World Wide Web Consortium (W3C): eXtensible Markup Language (XML). Internet: http://www.w3.org/XML/ (1997)