XRoaster: A Tool for Catalog Management on Middleware Databases Systems

Ana Mercedes Molina Vargas
Advisor: Manuel Rodríguez Martínez

Electrical and Computer Engineering Department
University of Puerto Rico, Mayagüez Campus
Mayagüez, Puerto Rico 00681
ana.mercedes@ece.uprm.edu

Abstract

Currently there are a few tools for catalog management, everything is done manually and require write code to change the global schema. XRoaster is a graphical tool for catalog management from integrated distributed data sources and is designed to generate XML documents which contain metadata about data sources. All the tasks for catalog management are integrated in this tool for use by the administrator. This application is going to improve the process of catalog management because it is fast and efficient, user friendly and it can handle all schema management tasks. This application does not require the administrator to write any code to do the management.

1. Introduction

A Schema is a logical framework on which to base the physical design of databases. A schema is an overall logical view of the relationships among the data elements in a database. A Database Middleware system is the software that helps diverse networked computer systems work together, thus promoting their interoperability.

Schema mapping is an important issue since it provides the mechanism by which the system extracts the required data from the data sources. Currently, however, there are few tools for catalog management and schema mapping management. Everything is done manually, and is necessary to write some ad-hoc code at the moment that is necessary for something to change in the global schema.

The catalog management is one of the most difficult problems for the administrators of database middleware systems. For this reason it is necessary one tool that interacts with the administrator and suggests possible actions to help the administrator to efficiently manage the catalog.

XRoaster uses the XML standard to provide a platform-independent framework that allows administrators to easily generate all the metadata necessary to register one or more data sources into the system. This application is going to be user friendly in order to improve the process.

In this report, we describe the principal characteristics of catalog management, and it is organized as follows: Section 2 presents a brief description of The Middleware System Architecture, The Catalog, The Schema Mapping and The Data Transformation. The issues of schema mapping are described in Section 3. In section 4 we describe The XML language and finally we have the description of Vb tool *XRoaster*.

2. Theory

When the information is integrated from different data sources, relate data maybe represented quite differently in each data source. Generally, each database is mapped into a single target representation and needed transformations in order to be added to a global schema. This transformation is changes the structure of the data [2].

In this section I review the work that is most related to this research effort.

2.1 Middleware

Middleware, is a layer of software between the servers and the applications. This software provides services such as identification, authentication, authorization, directories, and security.

Database Middleware systems are used to integrate collections of data sources distributed over a computer network [4].

Database Middleware is software that manages the communication between a client program and a database. For example, a Web server connected to a database can be considered middleware--the Web server sits between the client program (a Web browser) and a database. The middleware allows the database to be changed without necessarily affecting the client, and vice versa.

2.2 System Architecture of Middleware Systems Database.

The schema mapping is a set of definitions for all attributes matching all the columns in a database table or view

Typically, these types of systems follow an architecture centered around a data integration server, which provides client applications with a uniform view and access mechanism to the data available in each source [4].

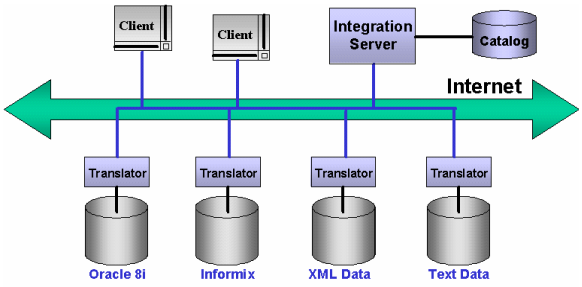

Figure 1. System Architecture

The Integration Server is the server that stores the data sets for a particular data site. Translator is the system that helps to provide a uniform access mechanism to a remote data source.

2.2.1 Catalog

The catalog contains metadata about views defined over the data sources. Metadata are data about data, data describing the structure, data elements, interrelationships, and other characteristics of a database [3].

This information is used to describe user defined data types, used defined operators, and any other relevant information such as selectivity of various operators.

To interpret the data of interest to users, whose applications need metadata. Therefore, whenever data is moved or transformed metadata must be involved. Metadata controls the movement by specifying such information as location, schedule, and source to target mappings.

2.3. Schema Mapping

Database Middleware systems, are used to integrate heterogeneous data sources dispersed over a computer network. In order to achieve data integration, the middleware system imposes a global data schema on top of the individual schema used by each source. Through this mechanism, the client applications are provided with a uniform view and uniform access interface to the data sets stored in each data source [5].

Global schema mapping defines the derivation of each integrated relation in a global schema from a set of related export relations.

Generally the local data is stored in tables, which have very different and complicated schemas (depending of the number of attributes), so it is necessary map each local schema into a much simpler global schema which sometimes contains different attributes. The result schema map can then used by the global schema.

2.4 Data Transformation

Views are an important means of reformatting data, especially for middleware, as the data resides in data sources. Converting from one data representation to another is time-consuming and labor intensive because in one common scenario an administrator of a database wants to add data from a new source and the data in the new source may no match the existing tables. Usually many applications today need information from several data sources, in which related data may be represented quite differently. Besides the new data may also be partially redundant with that in an existing database or it may be formatted differently [2].

For the schema mapping rules one or more data sets must be mapped onto a single target representation. Needed transformations may include schema transformations (changing the structure of the data) and data transformations.

3. Discussion

Schema Management is one of the most difficult problems faced by administrators of database middleware systems that are deployed over wide area environments such as the Internet [4].

We have developed an application framework for schema management; this is XRoaster – an XML-based tool for distributed catalog management that is fully integrated with the catalog and execution engine of the middleware system.

XRoaster is a tool designed to generate XML data containing metadata about data sources.

The goal of XRoaster is to aid the administrator to handle all schema management tasks, and not require him/her to write any specific software each time. With XRoaster the final user (that is the administrator) is going to use a graphical interface to browse the schema mapping, and can generate XML data containing the specification of schema.

This application easily creates the infrastructure needed to register data sources, user defined code, and schema mappings in database middleware systems.

4. XML Formats.

The Extensible Markup Language (XML), is a standard for data exchange over the Internet. XML is a markup language that is designed to encode the content in a document.

This arrangement is what makes XML documents machine readable, or self-describing, since applications can parse the XML document and find the tags enclosing the data they need to process [4].

Schema Management is the process of harvesting, storing and maintaining all the metadata that is necessary to implement a global schema of top of the local schemas used by the participating data sources.

The catalog contains the definitions of global tables (views) defined over the data sources, user defined data types and user defined operators. This metadata for each resource is specified in a XML document based technology.

In order to building the catalog, it is necessary build a XML encoding.

5. VB Tool

We developed a Visual Basic tool with a GUI, which has all the options to do the management process.

As part of schema management, an administrator is responsible for:

- Registering new data sources.
- Implementing a global schema that unifies the data sources.
- Registering new user defined code.
- Defining sets of schema mappings that transform the data from the local schemas at the sources into de global schema.

The application maps the data from the local schema into the global schema and also implements a schema mapping framework.

The GUI's help the administrator to create the schema mapping and the XML documents that contain the types of query capabilities of the source; this document is enabling to understand the catalog.

The principal characteristics of XRoaster are:

- It defines a new global schema that will be used to unify the different data sets.
- It creates a schema mapping between the local schema used at each data source and the global schema.
- It defines schema mappings from local data.
- It adds Local Table
- It adds Global table
- It adds new Type
- It adds new Operator

The current version of XRoaster is implemented using Visual Basic .Net. XRoaster uses ADODB .Net to connect with the database which contains the Global Schema and is implemented with a relational database. The user interacts with the application through a Graphical User Interface to execute the options and visualize their results.

The database used to create the global schema is implemented with Microsoft SQL Server 2000. Always the user can visualize the results, and some of the results can be an XML document that is going to be automatically generated.

References

[1] Bertino, Elisa. and Catania, Barbara. **Integrating XML and Databases**, IEEE Internet Computing, Volume: 5 Issue: 4, July-Aug. 2001
Page(s): 84 -88

[2] Hass L. M. **Transforming Heterogeneous Data with Database Middleware: Beyond Integration**. IEEE Internet Computing. 1999.

[3] O'Brien, James A. **Management Information Systems**. 1999

[4] Rodríguez Manuel, Roussopoulos Nick. **MOCHA: A Self-Extensible Database Middleware System for Distributed Data Sources**.1999

[5] Rodríguez Manuel, Roussopoulos Nick. **Automatic Deployment of Application-Specific Metada and Code in MOCHA**. 2000

[6] World Wide Consortium, XML Language Specification and related documents, at http://www.w3.org/TR/REC-xml

[7] Miller Reneé, Hass Laura, Hernandez Mauricio. **Schema Mapping as Query Discovery.** 2000.

[8] Lakshmanan Laks, Sadri Fereidoon, Subramanian Iyer. **SchemaSQL – A Language for interoperability in Relational Multi-database systems.** 2002

[9] Claypool Kajal, Rundensteiner Elke. **Flexible Database Transformations: The Serf approach.** 2001

[10] Bernstein Philip, Bergstraesser Thomas. **MetaData Support for Data Transformations Using Microsoft Repository.** 2001

[11] Hass L., Miller Reneé, Niswonger B., Tork Roth M., Schwarz P., Wimmers E. **Transforming Heterogeneous Data with Database Middleware: Beyond Integration.** 2001

[12] Carrer Marco, Jhosi Ashok, Lin Paul, Srivastava. **Metadata Transformation and Management with Oracle Intermedia.** 2001

[13] Orenstein Jack. **Supporting Retrievals and Updates in an Object/Relational Mapping Systems.** 2001