

**XROASTER: A TOOL FOR CATALOG MANAGEMENT ON
DATABASES MIDDLEWARE SYSTEMS**

By
Ana Mercedes Molina Vargas

A thesis submitted in partial fulfillment of the requirements for the Degree of

MASTER OF SCIENCES
IN
COMPUTER ENGINEERING
(Software)

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2003

Approved by:

Bienvenido Vélez, Ph.D
Member, Graduated Committee

Date

Néstor Rodríguez Rivera, Ph.D
Member, Graduated Committee

Date

Manuel Rodríguez Martínez, , Ph.D
President, Graduated Committee

Date

Paul Castillo, Ph.D.
Representative of Graduate Studies

Date

José L. Cruz Rivera, Ph.D.
Chairperson of the Department

Date

Abstract

Currently there are very few tools available for managing the catalog of a Database Middleware System. The system administrators of the Middleware System need to manually add all the metadata necessary to configure and maintain the catalog, making catalog maintenance a tedious task. In this thesis we present XRoaster, a visual tool to perform catalog management tasks in Database Middleware Systems. XRoaster is designed to generate its results in XML documents that contain metadata about the data sources. This makes it possible to easily share system metadata with other applications. XRoaster is a graphical, user friendly tool and that follows the rules of usability for the creation of Graphical Interfaces. One of the major features of XRoaster is its ability to guide the end-user in the administrative tasks: adding users, implementing global schemas, adding new sources of data, defining the required schema mapping rules. We argue that XRoaster provides a very robust framework for catalog management.

Resumen

Actualmente existen muy pocas herramientas disponibles para la administración del catálogo de un Sistema de Base de datos Middleware. Los administradores de sistema Middleware necesitan adicionar manualmente la metadata necesaria para configurar y mantener el catálogo, convirtiendo el mantenimiento del mismo una tarea tediosa. En esta tesis presentamos XRoaster, una herramienta visual para realizar las tareas de administración de catálogo en un sistema de Bases de Datos Middleware. XRoaster es una herramienta diseñada para generar los resultados en documentos XML que contienen la metadata de las fuentes de datos. Esto hace posible compartir fácilmente la metadata del sistema con otras aplicaciones. XRoaster es una herramienta gráfica y amigable con el usuario y que sigue las directrices y reglas de usabilidad para la creación de Interfaces Gráficas. Una de las características principales de XRoaster es la habilidad para guiar al usuario final en las tareas de administración: adicionar usuarios, implementar esquemas globales, adicionar nuevas fuentes de datos, definir las reglas necesarias de traducción de esquemas.

**© Copyright by
Ana Mercedes Molina Vargas
2003**

Dedication

To my parents and family, who offered me unconditional love and support throughout the course of these Master studies, and for their guidance, support, love and enthusiasm. Without these things this thesis could not have been possible.

Acknowledgement

I would like to thank my thesis advisor, Dr. Manuel Rodriguez Martinez for providing me with the concept for this thesis and supplying me ample resources so that it could all be done. I would also like to thank my graduate committee members: Dr. Néstor Rodriguez, Dr. Bienvenido Vélez, and Dr. Paul Castillo for their effort in helping to make this project a success.

My appreciation goes to the **Program for Research in Computing Information Systems and Engineering (PRECISE)** people for allowing me to use their research facilities during this work, thanks for all your support throughout this process.

Also I would to express my sincere appreciation for all the people that contributed to the successful development of this Thesis. The project could not have been implemented without the cooperation of the usability test people who participated in the study.

Thanks to my mother for her support and prayers to finish this thesis. I would like to acknowledge my father, brother, sisters and family, for following up my work. I cannot forget my lovely nephew Juan Camilo. Thanks to my friends and colleagues; I wish to express my appreciation for their time and moral support.

Finally, a very special thanks to Amilcar Y. Rodríguez, who always held out a positive view for me, who assisted me with several tasks in the preparation of the thesis, and who was the most consistently friendly and supportive person I have known throughout the almost three years of this undertaking.

Table of Contents

LIST OF TABLES _____	IX
LIST OF FIGURES _____	X
LIST OF ANNEXES _____	XI
1. INTRODUCTION _____	1
1.1 Motivation _____	1
1.2 Problem Statement _____	3
1.3 Outline of thesis _____	5
2 REVIEW OF LITERATURE AND RELATED WORK _____	6
2.1 System Architecture _____	6
2.2 Schema Mapping _____	8
2.3 Related Work in Visual Tools _____	11
3 METHODOLOGY _____	13
3.1 XRoaster Organization Overview _____	13
3.2 XRoaster Connectivity _____	14
3.3 Interactive Schema Integration in XRoaster _____	15
3.4 Schema Mappings Rules _____	21
3.5 Data Extraction _____	24
3.6 Schema Mapping Metadata _____	26
3.7 Usability Test _____	27
3.7.1 Test Purpose _____	28

3.7.2	Test Goals	28
3.7.3	User Profile	29
3.7.4	Test Methodology	29
3.7.4.1	Introduction or Welcome to the User	30
3.7.4.2	Forms of Data Collection.	30
3.7.4.3	Pre Training.	30
3.7.5	List of Tasks	31
3.7.6	Atmosphere	31
3.7.7	Criteria Test	32
4	DISCUSSION AND RESULTS	33
4.1	Data Collection	33
4.2	Usability Tests	33
4.2.1	Icon Evaluation	34
4.2.2	Objective Measures	35
4.2.3	Preference Measures	39
5	CONCLUSIONS AND FUTURE WORK	57
5.1	Conclusions of Research	57
5.2	Future Work	59
	BIBLIOGRAPHY	61

List of Tables

Table 1 Relational Table us_landsat	22
Table 2 Relational Table granule	22
Table 3 Global Table Granule	22
Table 4 Mappings between the source & target schemas (Granule -granule)	23
Table 5 Mappings between the source & target schemas (Granule / us_landsat)	23
Table 6 Tasks Description	31
Table 7 Results Icon Evaluation	34
Table 8 The number of completed tasks	35
Table 9 Attempts to make the task correctly	36
Table 10 Run time of the tasks	37
Table 11 Total Run time of the tasks	38
Table 12 Assertion 1 Score and Descriptive Statistics	40
Table 13 Assertion 2 Score and Descriptive Statistics	41
Table 14 Assertion 3 Score and Descriptive Statistics	42
Table 15 Assertion 4 Score and Descriptive Statistics	43
Table 16 Assertion 5 Score and Descriptive Statistics	44
Table 17 Assertion 6 Score and Descriptive Statistics	45
Table 18 Assertion 7 Score and Descriptive Statistics	46
Table 19 Assertion 8 Score and Descriptive Statistics	47
Table 20 Assertion 9 Score and Descriptive Statistics	48
Table 21 Assertion 10 Score and Descriptive Statistics	49
Table 22 Assertion 11 Score and Descriptive Statistics	50
Table 23 Assertion 12 Score and Descriptive Statistics	51
Table 24 Assertion 13 Score and Descriptive Statistics	52
Table 25 Assertion 14 Score and Descriptive Statistics	53
Table 26 Assertion 15 Score and Descriptive Statistics	54
Table 27 Assertion 16 Score and Descriptive Statistics	55

List of Figures

Figure 1 System Architecture	6
Figure 2 XRoaster 's Connection	15
Figure 3 XRoaster's Principal Form	16
Figure 4 Local Schema	18
Figure 5 Create Global Schema	18
Figure 6 Relations between tables	19
Figure 7 Schema Mapping	19
Figure 8 Expression Builder	20
Figure 9 Table granule's XML schema	26
Figure 10 Schema Mapping Metadata	27
Figure 11 Total Run Time of the tasks	39

List of Annexes

Annex A Letter Introduction to the user _____	62
Annex B Forms Data Collection _____	63
Annex C Presentation of XRoaster _____	65
Annex D Instructions Test _____	66
Annex E Results Form _____	70
Annex F Questionnaire User Satisfaction _____	71
Annex G Icon Evaluation Form _____	73

1. Introduction

1.1 Motivation

Currently, it is very common to find the companies and organizations with information system that are not integrated, and which cannot interoperate. Often these data sets are semantically related between them, but contained in systems that do not allow for their treatment and data access in a uniform way. Besides the data produced in the organization, there, exists external data that are important for the company but which are not generated locally. These sources of data can be diverse, and very important, for example stock information from the Internet.

The mission of the process of integration of information in an organization consists of discovering, storing and maintaining the existing relations between its data, in order to improve the different processes that are carried out in the organization (process, production, decision making, etc.) Major system integration is being performed in many organizations. From a technological perspective, system integration is the melding of divergent and often incompatible technologies, applications, data, and communications into a uniform information technology architecture and functional working structure [Mische00]. Such integration involves: 1) Data sharing and exchange, 2) Control coordination and 3) The need to facilitate future integration with additional related tools [Shaosong02].

Also, in recent years, there has been a tremendous proliferation of databases in the work place, dominated by relational database systems. An emerging need for sharing data and programs across the different databases has motivated the need for Multi-database systems, also referred as Heterogeneous Database Systems, Federated Database Systems or Database Middleware Systems. These systems are capable of operating over a distributed network (e.g. Internet) and encompassing a heterogeneous mix of computers, operating systems, and communications links [Molina03b]. Certainly, Database Middleware Systems have become a highly desirable and important technology [O'Brien99].

In a broad sense, a Middleware System is a layer of software that sits between servers and client applications. This software provides services such as identification, authentication, authorization, directories, and security [Rodríguez99a]. Thus, Middleware is software that manages the communication between a client program and a server. Middleware is also used to hide the specifics of server connectivity from the application. In this way, the servers can be changed without affecting the clients. 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.

A Database Middleware System is a Middleware System that also has built-in distributed query processing capabilities and schema transformation routines. 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 schemas used by each source being integrated. In this way, the client applications

are provided with a uniform view and uniform access interface to the data sets stored by each data source [Rodríguez00b].

For a given application, the data that need to be integrated could be represented quite differently in each data source. Therefore, when records of information are integrated from different data sources, each local record is mapped into a single target representation (a global schema) to be handled by the target application. This process is called Schema Mapping. To accomplish these mappings, it is necessary to have rules that define the transformations that convert the local record into a global record in the global schema. Notice that these transformations can change the internal structure of the data [Hass01]. Converting from one data representation to another is an intensive, difficult and error-prone labor, particularly because, in a common scenario, an administrator might want to add data from a new source and the necessary transformation might be non-trivial. Reasons for this complication include: different data formats, lack of a well-defined schema (e.g. flat files), or partially redundant information [Ling01].

1.2 Problem Statement

Clearly, schema mapping is an important issue since it provides the mechanism by which the system extracts the required data from the data sources. The catalog of the Middleware System is designed to store metadata for schema mapping. It includes information about data sources, local tables, global tables, data types, query operators, and schema mapping rules. One of the main problems faced by system administrators is that there are very few tools for catalog management and schema mapping management. Most tasks are done

manually, and it is often necessary to write some ad-hoc code to make any change in the global schema. Thus, 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 [Rodríguez99a]. For this reason it is necessary to have a tool that interacts with the administrator and suggests possible actions to help the administrator to efficiently manage the catalog.

In this thesis we present a framework for schema management and its realization with 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 an XML-based tool that is designed to generate XML data containing metadata about data sources. 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.

The main purpose of XRoaster is to assist the administrator in the handling of all schema management tasks, and require him/her to write as little specific code as possible each time a new source is integrated. With XRoaster the administrator is going to use a graphical interface to browse the catalog, and define schema mapping rules. XRoaster can generate XML data containing the specification of schema [Larson00] in order to make it available to other applications. XRoaster can be used to easily create the infrastructure needed to register data sources, user defined code, and schema mappings in Database Middleware Systems. The execution engine of the Database Middleware System can then use these rules, and convert them into a query plan data structure that performs the mapping of data from local schemas to global schemas.

As part of our work, a usability test has been performed, which describes the XRoaster quality from the point of view of the user. The user interacts with XRoaster and the systems through user interfaces, which allow the user to understand and use the tool. The aim of the test is to collect data in three types of ways: 1) The icon evaluation, where the user can describe what they understand the icon to mean, 2) Objectives (performance), are related to these performance measures: a) The number of completed tasks, b) Number of Attempts to complete the task correctly, and c) Run time of the tasks. 3) Finally, the subjective measures are collected (Preference): easy-to-use, utility of the application and user satisfaction. Our usability tests present evidence of a tendency of positive feedback and very good user acceptance of the tools.

1.3 Outline of thesis

The remainder of this thesis is organized as follows. Section 2 explores the Catalog Management on Middleware Database Systems issues and Related Work. The specific issues associated with the XRoaster System are described in Section 3: the experimental design and the methodology used for the development of the work are presented, as well as the usability tests plan. Section 4 presents the results obtained and their analysis. Section 5 concludes this thesis with the contributions of the research and discussion about future work. Finally, the Bibliography and the respective Annexes are presented.

2 Review of Literature and Related Work

2.1 System Architecture

The figure 1 depicts the typical architecture of a Database Middleware System. Typically, these types of systems follow an architecture centered on a data *Integration Server*, which provides client applications with a uniform view and access mechanism to the data available in each source [Molina03a].

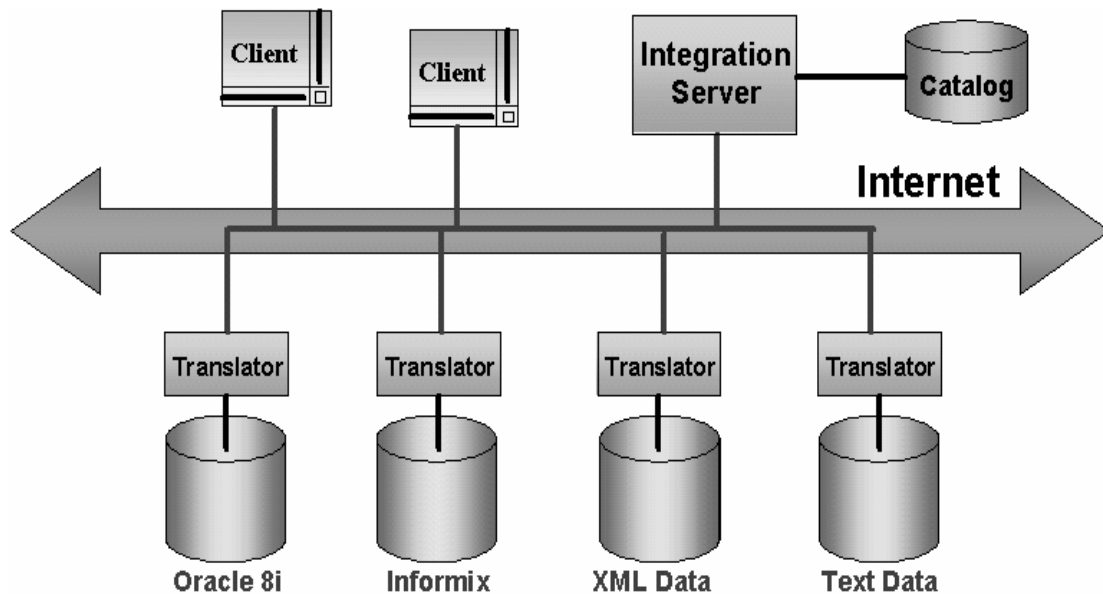


Figure 1 System Architecture

The *Data Sources Server* is the server that stores the data sets for a particular data site. Examples of these include Oracle, Informix, XML engines, and custom-built file systems. The *Translator* is the system that helps to provide a uniform access mechanism to a remote data source. Moreover, each translator contains

the code that perform the mapping of the local schemas to the global schema imposed by the middleware system.

As shown in Figure 1, the catalog is related directly with the Integration Server. The catalog contains metadata about views defined over the data sources. These views defined a global schema that makes all the sources appear as if they had the same structure. In addition, the catalog has information about data types, data describing the system structure, data elements, interrelationships, and other characteristics of a database [Hass01]. This information is also used to describe user-defined data types, used-defined operators, and any other relevant information such as selectivity factors of various query predicates.

In order to interpret the data of interest to users, the applications need metadata about it. Therefore, whenever data is moved or transformed, metadata must be involved. Metadata controls the action by specifying such information as location, schedule, and source-to-target mappings. Metadata needs to be exchanged together with the data so that further work can be done on the data after the exchange. The information is maintained in the catalog because, like any kind of data, the metadata requires a place for storage. It also requires an information model (i.e. schema) that describes the contents of the metadata [Bernstein01]. In our approach, we store the metadata in XML format.

Generally the local data is stored in tables, which have very different and complicated schemas (depending on the number of attributes), so is necessary map each local schema into a much simpler global schema which sometime contains different attributes. The resulting schema map can then used to materialize the data at the global schema level. In order to realize the schemas

and mappings, an administrator needs to fill the catalog with the metadata about:

Local Tables: The metadata for local tables must be available to the system in order to enable query processing. For each table, metadata are generated and added to the catalog. This information will enable database access, retrieval of tuples (data records) and reading each base column value into a middleware type for a global column.

Global Tables: The metadata added to the catalog resembles that added for local tables.

Schema Mappings Rules: These rules establish the correspondence between the columns in the global schema and those in the local schemas. They are used to generate a query plan data structure that implements the logic to convert data from the local to the global schema.

2.2 Schema Mapping

The process called Schema Mapping consists in converting from one data representation to another. Schema mapping is an intensive, difficult and error-prone labor, particularly because, in a common scenario, an administrator might want to add data from a new source and the necessary transformation might be non-trivial.

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. Global schema mapping defines the derivation of

each integrated relation in a global schema from a set of related export relations. That is, it provides the mechanism to transform the data from the local representations (local schema) to the global representations (global schema) [Molina03a]. When integrating data from data sources into a unified relational database, there are three issues to be solved [Larson00]: (1) Detect and resolve structural inconsistencies, (2) Detect and resolve value inconsistencies, (3) Balance the trade-off between the cost of maintaining an accurate database with the cost of mistakes due to inaccurate data.

For a given application, the data that need to be integrated could be represented quite differently in each data source, this is Structural inconsistency. Types of structural inconsistencies include naming issues, inconsistent coding structures, inconsistent data formats and unneeded or missing data [Shaosong02]. For example two similar data items that there are in different local data sources may represent similar information and two different items can have the same name.

Inconsistent coding structures means that the values used to represent information about the same data may differ between two data items. Data elements from different databases frequently use different data formats. For example, a date may be represented as mm/dd/yy in a one local data source and in another may be represented as yy/mm/dd. When records of information are integrated from different data sources, each record is mapped into a single target representation (a global schema). To accomplish these mappings, it is necessary to have rules that define the transformations that convert the record into a global schema. Notice that these transformations can change the internal structure of the data [Molina03a].

For data sharing or data integration we use this approach: 1) designing the target schema (Global Schema - data elements needed in the target schema must be determined, and 2) constructing Schema Mapping rules - mappings between the source and target schemas must be defined.

For integrating local schema into global schema, we use the schema mapping rules. The different data schemas resulting from different data modeling in each local data source should be mapped with one another to obtain an integrated data schema (Global Schema) for the shared information, even though the data formats are exchangeable. To perform the process needed to integrate local data sources in a global data schema, mappings must be established between the data schema of a local database and the global data schema.

One or more local data sets must be mapped into a single target global representation for the schema mapping rules. Needed transformations may include schema transformations (changing the structure of the data) and data transformations (changing the data types).

To establish the schema mapping rules we use the following approach introduced by Larson [Larson00]. First, classify the relationship between two source tables A and B into four types: 1) Logical containment - if each row in A always corresponds to a unique row in B, then B logically contains A, 2) Logical equivalence - if each row in A always corresponds to a unique row of B and each row in B always corresponds to a unique row in A, then B is logically equivalent to A, 3) Logical overlap - if some of the rows in A correspond to some of the rows in B and some of the rows in B correspond to some of rows in A, and 4) Logically disjoint - if none the rows in A correspond to any of rows in B and none of the

rows in B correspond to any of the rows in A, then A and B are logically disjointed. If two source tables are logically disjointed, they should not be integrated into a single table. Two logical equivalent tables should be integrated into a single target table that contains the union of the columns from the two source tables with columns suitably renamed. Multiple target tables should be created to replace the two source tables to be integrated. One target table contains the columns and rows common to both source tables being integrated. The other additional targets table contains the key from the source table and the columns from the source table not common to the two source tables to be integrated.

Schema integration techniques typically distinguish two key tasks: creation of the integrated schema and creation of queries (mappings) between schemas [Miller2000]. For these two tasks that the user have to use his/her experience and knowledge about the particular application domain to be used with the system.

2.3 Related Work in Visual Tools

We have made a visual tool that groups the tasks that a user must make for catalog management, and that at the moment are centered in the ad-hoc codification. This makes it a tedious, repetitive and time consuming task. Other related tools provide a way to generate the schema mapping rules, like Clio, a tool that creates mappings between two data representations with user input [Orenstein99],[Molina03a].

Catalog allocation schemas like the tool GOCAS (Group Oriented Catalog Allocation Schema) [Cho97], and tools that are tightly integrated with an easy

report generation, can readily display the query result. These requests are processed by a metadata that translates them into SQL queries, like ad hoc query tools, including Esperant and Impromptu [Ling01]. These tools do not group all the tasks that imply catalog management of Middleware Database Systems like: 1) define a new global schema that is used to unify the local schemas, 2) create a schema mapping between the local schema used in each source of data and the global schema, 3) add neither local tables or global tables, etc., and 4) also to generate all the results in a XML document to be distributed in the network.

3 Methodology

3.1 XRoaster Organization Overview

We developed XRoaster - a Visual tool with a GUI, which has all the options to do the management processes related with schema mapping. The goal is to handle most schema management tasks automatically, or semi-automatically. In addition, we would like to minimize the amount of code that the administrators need to write to perform schema mapping operations. Finally, XRoaster should be able to make suggestions about possible mapping, or possible inconsistencies or errors in the mapping being attempted. However, these latter two features are outside the scope of this thesis, and are suggested as future work.

As part of schema management, an administrator is responsible to: 1) register new data sources and user defined code, 2) implement a global schema that unifies the data sources, and 3) define a set of schema mappings that transform the data from the local schemas at the sources into the global schema.

XRoaster maps the data from the local schema into the global schema and also implements a schema mapping framework. The GUI helps the administrator to create the schema mapping and the XML documents that contain the types of query capabilities of the source; this document will enable the understanding, processing and exchange of the metadata held in the catalog [Molina03b].

The principal operations supported by XRoaster are:

- Define a new global schema that will be used to unify the different data sets.
- Create a schema mapping between the local schema used at each data source and the global schema.
- Define schema mappings from local data.
- Add Local Tables, Add Global tables
- Add new Types, Add new Operators

3.2 XRoaster Connectivity

XRoaster connects to the server where the local data base is located to be able to extract the necessary information from its local schema. For this purpose, XRoaster displays a window where the user has the opportunity to choose the server that he/she wants to connect. After making contact, all the accessible data bases that are held in that server are shown, together with the database engine which generated those databases. This is shown in figure 2.

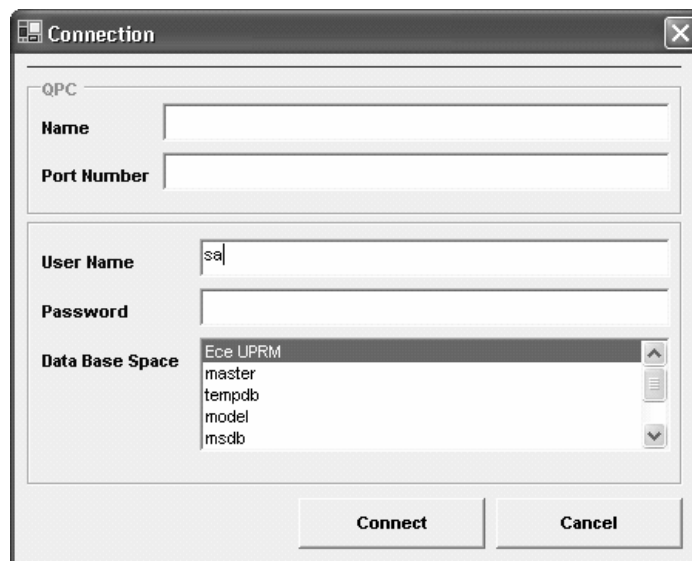


Figure 2 XRoaster 's Connection

The current version of XRoaster is implemented using Visual Basic .NET. XRoaster uses ADODB .NET to connect with both the local databases and the database that contains the Global Schema. The global schema is represented in two ways: as a relation and in an XML document. In this prototype, the database used to create the global schema is implemented with Microsoft SQL Server 2000. [Molina03b]

From ADODB .NET, XRoaster uses two providers to access data – Open Database Connectivity (ODBC) and Object Linking and Embedding Database (OLE DB). The ODBC .Net Data Provider works as a bridge toward an ODBC source; we use ODBC to carry out the SQL commands to store and retrieve data from the local database connected to the server that the user want to connect, as well as the global database. There are two parts to ODBC – core functionality and the extended SQL functions [Shaosong02].

OLE DB .NET data Provider is a set of interfaces and methods that helps us to access persistent data stores like the local and global databases. The OLE DB provides a common API, which can be used to access different sorts of data. This data could be in tables, databases, files etc. The data could be relational data; it could be data in a flat file or even data in spreadsheets or hierarchical file systems [Shaosong02]. The user of XRoaster could access data in XML format easily thanks to OLE DB, which acts supplementary to ODBC.

3.3 Interactive Schema Integration in XRoaster

The user interacts with the application through a Graphical User Interface to execute the options and visualize their results. Some of the results can be XML

metadata documents that are going to be automatically generated. XRoaster maps the data from the local schema into the global schema and also implements a schema mapping framework. The GUI helps the administrator to create the schema mapping and the XML documents that contain the types of query capabilities of the source. This document will enable the understanding, processing and exchange of the metadata.

Figure 3 shows the principal form in XRoaster, where there are the principal options and principal menu. You can see that the user is going to find the several menu options to execute what he/she want. Other mechanisms that the user can used to perform administrative tasks are the shortcuts.

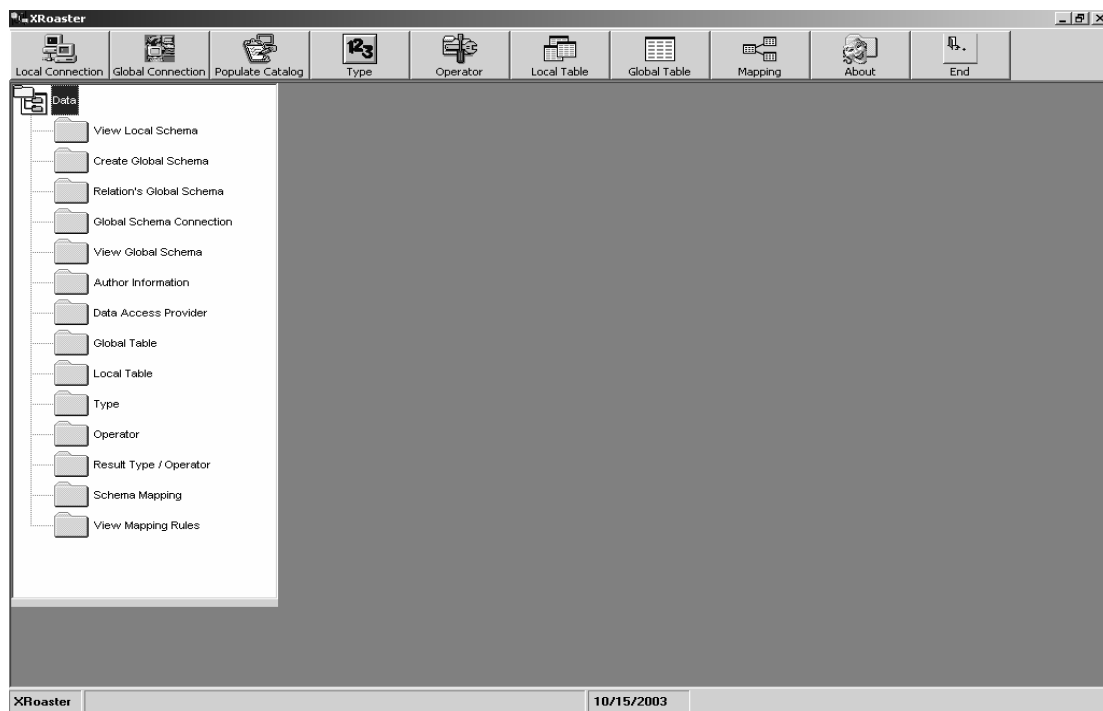


Figure 3 XRoaster's Principal Form

The Schema Integration aims at the resolution of the heterogeneity of data schema that comes from conceptual heterogeneity of data models, and different

data understanding and modeling in local databases. Because an integrated, global data (shared database) schema has been developed, using mappings between local data schemas and global schema, each heterogeneous local data schema can be transformed. A data model is a collection of high-level data description that hides many low-level storage details [Orenstein99].

We propose the following tasks to perform the process of Schema Integration:

- 1) Explore data schema of each local data source (Figure 4).
- 2) Compare all data schemas (called “local schema”).
- 3) Create an integrated global schema. (Figure 5 and Figure 6)
- 4) Generate schema mapping rules between local schema and global schema (Figure 7, See Page 20).

1) Explore data schema of each local data source.

Once the local schema has been defined in the local source of data, XRoaster connects to the server where the local data base is located to be able to extract the necessary information of its local schema. After the user has chosen the database, the application automatically reads the schema defined for this data source and displays the metadata on the screen so that the user can easily understand the distribution of the information. It shows all the tables on this database, and can also unfold the pertinent information to each table doing double click on it. This is shown in figure 4.

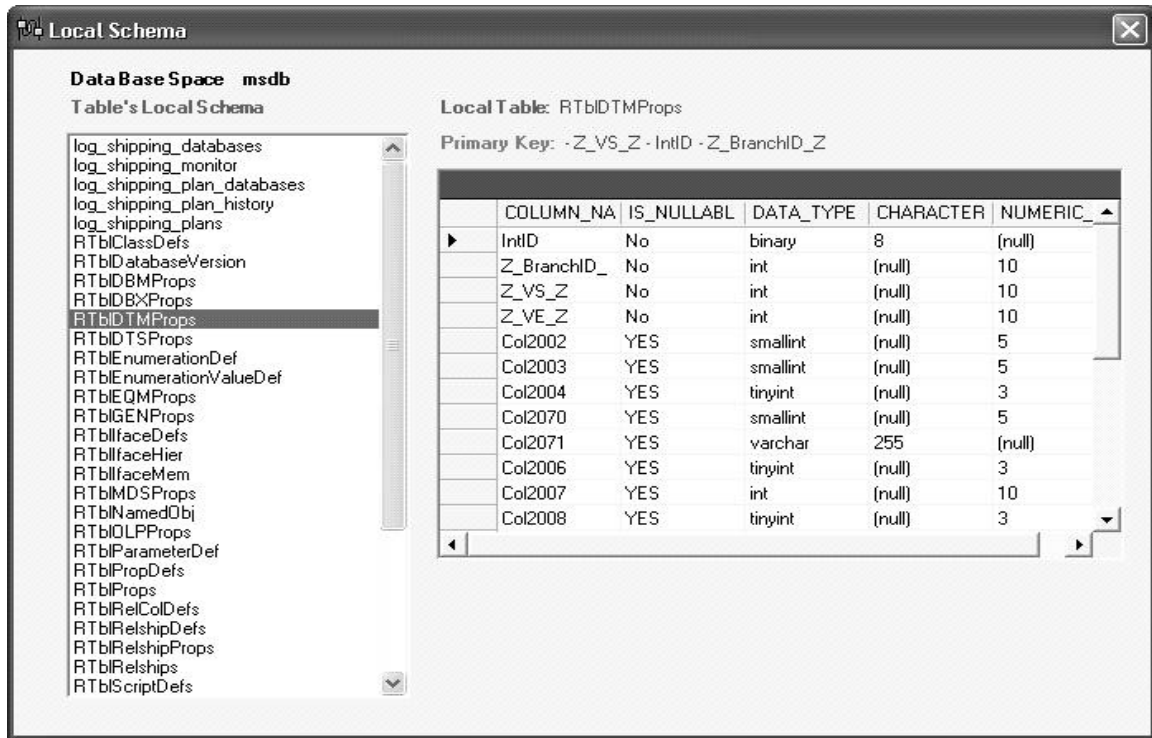


Figure 4 Local Schema

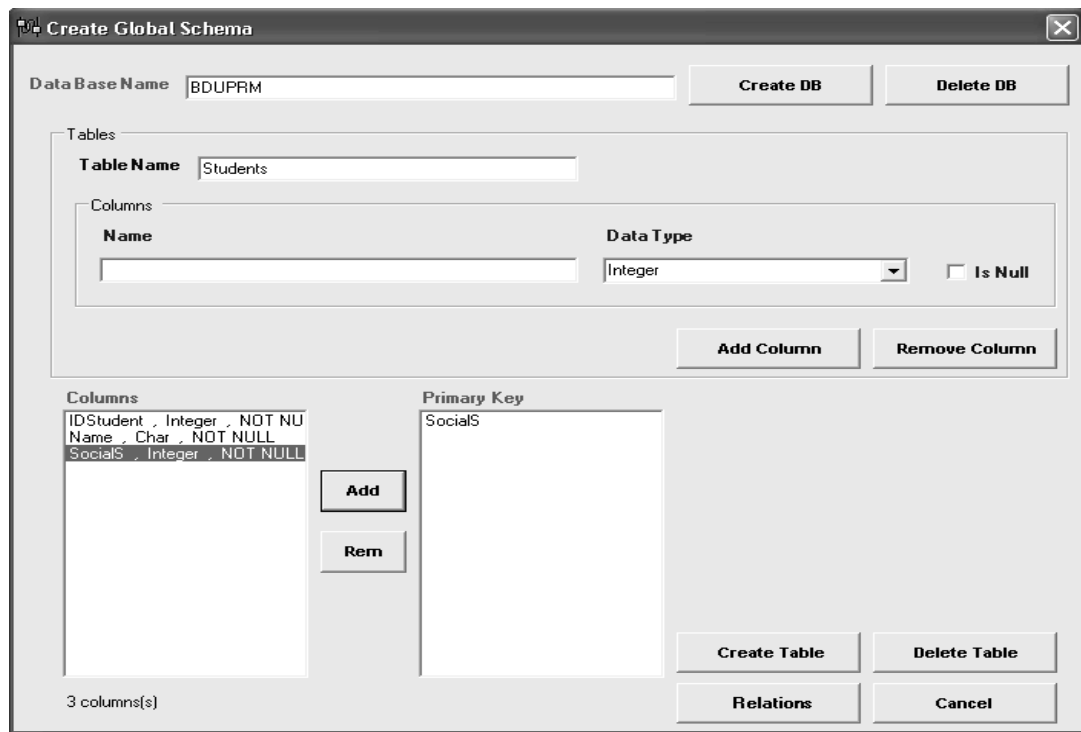


Figure 5 Create Global Schema

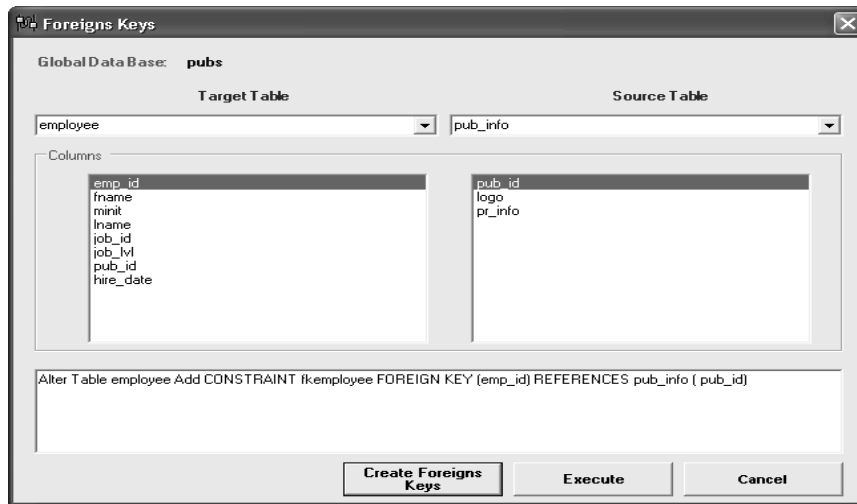


Figure 6 Relations between tables

After defining which are going to be the local and global schemes respectively, the definition of the schema mapping rules begins. These rules will allow to unify the local schema with the global schema. The prototype of the graphical interface can be observed in figure 7.

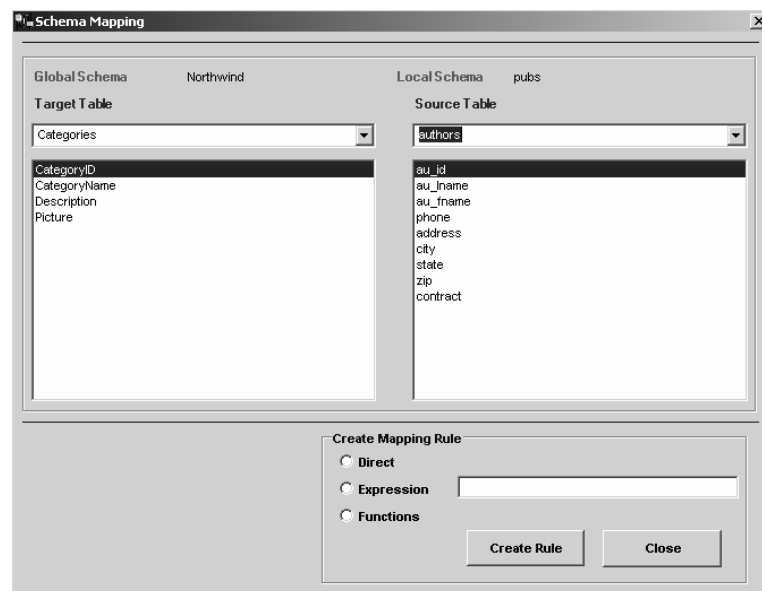


Figure 7 Schema Mapping

The user can find both the list of tables from the Global Schema and the list of tables from Local Schema. After he/she selects the table, the columns appear in a pannel below the table name.

The user has the option of Functions to perform the column transformation between tables. And in the screen appear the Expression Builder, Figure 8.

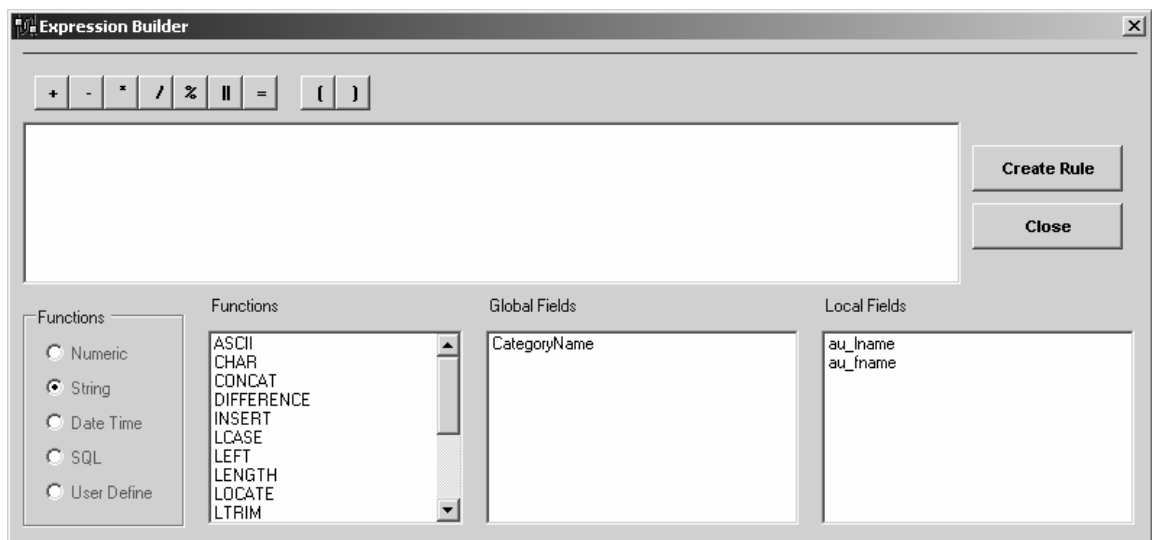


Figure 8 Expression Builder

The expression builder is a general purpose tool that helps the user to easily construct the mapping rules. The interface provides a hierarchical view of all support functions like Numeric, String, Date Time, SQL Functions and User Define functions, and current columns that the user selected in the back interface. Any of these elements can be selected to create complex expressions that are going to be the mapping rules.

At this moment it has been possible to see that the user will interact with graphical interfaces that facilitates the work to the user. This GUI interaction will render a more comprehensible structure of the schemas that he/she wants to unify.

The schema mapping becomes a processes of choosing the options with mouse's click so that the administrator creates the mechanism by which the middleware system extracts the required data of the data sources.

3.4 Schema Mappings Rules

XRoaster minimize the amount of code that the administrators need to write to perform schema mapping operations. The only required functions are those that convert custom types to the middleware types, as well as user-defined operators. Beside all the tasks are grouped in the tool, the user does not have to be interacting with several tools. He/she just simply need to click on the desired option of the menu to make the desired task.

We define the mapping rules at relational table level. That is, we have a set of mapping rules to map local columns into global tables.

Suppose that Tables `us_landsat` and `granule` (Local Sources Tables) are to be integrated into Global Table `Granule` (Global Table) in the Global Database. Their schemas are shown in Table 1, 2 and 3 (see tables 2 and 3 on page 22):

Table 1 Relational Table us_landsat

us_landsat	
Attribute	Type
Project_id	Integer
path_num	Integer
row_num	Integer
acq_date	date
data_path	Varchar
Dirname	Varchar
Filename_suf	Varchar
file_type	Varchar
bounding_box	Geobox
loc_desc	Varchar

Table 2 Relational Table granule

granule	
Attribute	Type
Granule_id	Integer
project_id	Integer
Granule_size	Integer
Bound	Poly
Path	Integer
granule.row	Smallfloat
start_date_time	Datetime
end_date_time	Datetime
Description	Varchar

Table 3 Global Table Granule

Granule	
Attribute	Type
GranuleID	Integer
ProjectID	Integer
GranuleSize	Integer
Bounds	MIQuadPoly
WRSPath	Integer
WRSRow	Integer
GranuleDate	Date
Description	String
PreviewURL	String

To integrate the source tables into a global table, the structural relationships between the source and target schemas should be identified. Unit inconsistency should also be detected and resolved. The mappings graphical representations between source and target schemas are shown in Tables 4 and 5.

Table 4 Mappings between the source & target schemas (Granule -granule)

Granule		granule
<i>Attribute</i>		<i>Attribute</i>
GranuleID	_____	granule_id
ProjectID	_____	project_id
GranuleSize	_____	granule_size
Bounds	_____	Bound
WRSPath	_____	Path
WRSRow	_____	granule.row
GranuleDate	_____	start_date_time
Description	_____	end_date_time
PreviewURL	_____	Description

Table 5 Mappings between the source & target schemas (Granule / us_landsat)

Granule		us_landsat
<i>Attribute</i>		<i>Attribute</i>
GranuleID		project_id
ProjectID	_____	path_num
GranuleSize		row_num
Bounds	_____	acq_date
WRSPath	_____	data_path
WRSRow	_____	Dirname
GranuleDate		Filename_suf
Description		file_type
PreviewURL		bounding_box

Global fields that have no counterparts on the local tables are set to default values, or to values that the user must specify by hand.

XML encoding software, which often supports data formatting services, can take the result of a query, expressed in the catalog query language, and encode the resulting data in an XML document to be transferred over the network [Bertino01], [Carrer01]. The query language is a specialized language to the catalog for writing queries about this information [Ramakrishnan03].

3.5 Data Extraction

To support data extraction, it is necessary to define a mapping between XML documents and the catalog data model.

A catalog data model is a collection of high-level data description that hides many low-level storage details. A database administrator has to define the metadata to be stored in terms of a data model [Orenstein99].

Using XML, we obtain Table granule's XML schema and us_landsat's Schema, in the same way, we have Table Granule's XML schema. Figure 9 depicts an XML generated by XRoaster, which shows the description of the local table granule. As we can see, the document describes details such a table name, cardinality and column information. In particular, column information is quite detailed, indicating: column name, column data type, number unique values currently stored, min and max values, and the size in bytes of the data types. This information can be used by the system to estimate result sizes and allocate resources accordingly.

```

<XRoaster:Table> granule </XRoaster:Table>
< XRoaster:Cardinality> 1418 </XRoaster:Cardinality>
< XRoaster:Columns>
<Seq>
  <li parseType = "resource">
    <XRoaster:Column> granule_id </XRoaster:Column>
    < XRoaster:Type> integer </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 1.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 1000.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 4 </XRoaster:AvgSize>
  </li>
  <li parseType = "resource">
    <XRoaster:Column> project_id </XRoaster:Column>
    <XRoaster:Type> integer </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 1.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 1000.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 4 </XRoaster:AvgSize>
  </li>
  <li parseType = "resource">
    <XRoaster:Column> granule_size </XRoaster:Column>
    <XRoaster:Type> integer </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 1.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 1000.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 4 </XRoaster:AvgSize>
  </li>
  <li parseType = "resource">
    <XRoaster:Column> bound </XRoaster:Column>
    <XRoaster:Type> poly </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 0.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 0.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 33 </XRoaster:AvgSize>
  </li>
  <li parseType = "resource">
    <XRoaster:Column> path </XRoaster:Column>
    <XRoaster:Type> integer </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 1.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 1000.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 4 </XRoaster:AvgSize>
    <XRoaster:URI>
  </li>
  <li parseType = "resource">
    <XRoaster:Column> granule.row </XRoaster:Column>
    <XRoaster:Type> smallfloat </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 1.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 1000.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 4 </XRoaster:AvgSize>
  </li>

```

```

    <XRoaster:URI>
  </li>
  <li parseType = "resource">
    <XRoaster:Column> start_date_time </XRoaster:Column>
    <XRoaster:Type> datetime </XRoaster:Type>
    <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
    <XRoaster:MinVal> 0.0 </XRoaster:MinVal>
    <XRoaster:MaxVal> 0.0 </XRoaster:MaxVal>
    <XRoaster:AvgSize> 8 </XRoaster:AvgSize>
    <XRoaster:URI>
  </li>
<li parseType = "resource">
  <XRoaster:Column> end_date_time </XRoaster:Column>
  <XRoaster:Type> datetime </XRoaster:Type>
  <XRoaster:NoUVals> 0 </XRoaster:NoUVals>
  <XRoaster:MinVal> 0.0 </XRoaster:MinVal>
  <XRoaster:MaxVal> 0.0 </XRoaster:MaxVal>
  <XRoaster:AvgSize> 8 </XRoaster:AvgSize>
  <XRoaster:URI>
  </Seq>
</XRoaster:Columns>
</Description>

```

Figure 9 Table granule's XML schema

3.6 Schema Mapping Metadata

There are various implementation methods for the representation and manipulation of these mapping relationships. One representation method is to store the XML mapping information into a table in the relational database. We chose the store mapping information into a table in relational database because Relational Database is faster to manipulate (e.g., select, project, join).

In the next figure, we present a representative XRoaster window in order to manipulate the schema mapping metadata.

Tables						
id Table	Name	Number Columns	Data Base	URI		
382	Categories	4	Northwind	(null)	(r	
383	CustomerCustomerDemo	2	Northwind	(null)	(r	
384	CustomerDemographics	2	Northwind	(null)	(r	
385	Customers	11	Northwind	(null)	(r	
386	Employees	18	Northwind	(null)	(r	
387	EmployeeTerritories	2	Northwind	(null)	(r	
388	Order Details	5	Northwind	(null)	(r	
389	Orders	14	Northwind	(null)	(r	
390	Products	10	Northwind	(null)	(r	
391	Region	2	Northwind	(null)	(r	
392	Shippers	3	Northwind	(null)	(r	
393	Suppliers	12	Northwind	(null)	(r	
394	Territories	3	Northwind	(null)	(r	

Columns							
Name Columns	Type Inf	Num Unique Value	Size	Min value	Max Value	URI	
CategoryID	(null)	(null)	(null)	(null)	(null)	(null)	(null)
CategoryName	(null)	(null)	(null)	(null)	(null)	(null)	(null)
Description	(null)	(null)	(null)	(null)	(null)	(null)	(null)
Picture	(null)	(null)	(null)	(null)	(null)	(null)	(null)
*							

Figure 10 Schema Mapping Metadata

3.7 Usability Test

Usability describes the quality of products and systems from the point of view of the human who uses them. Users interact with products and systems through user interfaces, which enable the user to understand and operate systems and products. According to Nielsen [Nielsen93], usability is a broad concept that refers to:

- Learnability - the easiness of learning to use a system.
- Efficiency of use - how efficiently it can be used.
- Memorability – the ease of constant use.
- Few and non-catastrophic errors - frequency of the errors.
- Subjective satisfaction - how pleasant it is to use.

A usability test is a concrete and objective measurement of the usability of a tool or system taken from users. This consists of assigning a series of tasks to a group of users with certain characteristics, so that it is possible to evaluate the impact of variables previously defined (for example the run time of tasks). This evaluation makes it possible to compare the behavior of different groups, or of the specific values of the established parameters by means of statistical analysis.

3.7.1 Test Purpose

The XRoaster application has been created in the department of Electrical and Computer Engineering, at the University of Puerto Rico, for catalog management of Middleware DataBase Systems. One of the major features of XRoaster is its ability to guide the end-user in the administrative tasks: adding users; implementing global schemas; adding new sources of data; and defining the required schema mapping rules. We argue that XRoaster provides a very robust framework for catalog management. This system is directed basically to one user type: Administrators of the Catalog of a Middleware System. The intention of these test are to determine the possible errors that can face the XRoaster's user when interacting with the tool. This was made by interface evaluation in the understanding and handling, on the part of the users, of each one of their menus.

3.7.2 Test Goals

The objectives of the usability test were:

1. Determine the capacity of the XRoaster system to help users complete their tasks through the use of the graphical menus.

2. Establish if the users understand the functionality of the mentioned menus.
3. Determine problems of navigation in the XRoaster Tool.
4. Evaluate user satisfaction with respect to the system.
5. Icons Evaluation.

3.7.3 User Profile

The number of users needed to make the test are 10 people. The main characteristics for these users are:

- To have a knowledge of data bases, data base management, and catalog data bases.
- To have experience in the handling of computers using Windows XP.

We also wanted do an icons evaluation. For this we needed 16 people who have a knowledge of data bases, in order to develop the test. We argue that this sample can be used to notice the trends in the usability of the tools. However, a more complete study with a larger number of users is need. Unfortunately, we did not have an abundant pools of expert users to carry this task. Making a more complete examination of XRoaster is part of our future work.

3.7.4 Test Methodology

- 3.7.4.1 **Introduction or Welcome to the User.** The introduction or welcome to the user will occur with a letter. (Annexe 1. Welcome Letter).
- 3.7.4.2 **Forms of Data Collection.** The design of the forms used can be observed in the annexes. In the first column is the tasks listing (12 tasks) and in the following columns are placed the evaluation variables mentioned previously. (Annexe 2. Forms of Data Collection)
- 3.7.4.3 **Pre Training.** For the user group, a small presentation of the XRoaster tool was produced as an introduction in order to understand XRoaster's functionality. We then presented a tutorial to the users with a specific XRoaster example. (Annexe 3. Presentation of XRoaster.)
- 3.7.4.4 **Instructions and Design of The Test.** The instructions corresponding to the user which contains a description of the tasks to follow.
- 3.7.4.5 **Questionnaires after The Test.** When the user finishes the test, a questionnaire was given to him/her in order for him/her to subjectively evaluate the system. (Annexe 4. Questionnaire User Satisfaction).
- 3.7.4.6 **Icon Evaluation.** For the user group, a small presentation of the XRoaster tool was created as an introduction, in order to understand XRoaster functionality. Then we gave a form to the user in order for him/her to write a brief description about the icon (Annexe 5. Icon Evaluation form).

3.7.5 List of Tasks

- 3.7.5.1 Presumptions. As the usability evaluation is for determining the errors in the interaction with the tool, any presumptions by the user occur by chance.
- 3.7.5.2 Task Profile. Now we present the tasks that the users will have to perform.

Table 6 Tasks Description

NUMBER TASK	DESCRIPTION
1	ENTER TO XROASTER
2	VISUALIZE LOCAL SCHEME
3	CONNECT TO GLOBAL SCHEME
4	VISUALIZE GLOBAL SCHEME.
5	ADD NEW AUTHOR AND DAP.
6	POPULATE CATALOG.
7	ENTER METADATA FOR GLOBAL TABLE.
8	ENTER METADATA FOR LOCAL TABLE.
9	CREATE DATA TYPE AND FUNCTION.
10	CREATE A DIRECT MAPPING RULE AND EXPRESSION.
11	CREATE A MAPPING RULE USING FUNCTIONS.
12	VISUALIZE MAPPING RULES.

3.7.6 Atmosphere

The test was developed in the workplace (University of Puerto Rico, Mayaguez Campus) of each one of the user's, so that they would feel more comfortable and confident. A personal computer is used to perform the test, using the platform of Windows XP and with the applications XRoaster and SQLServer 2000 installed.

3.7.7 Criteria Test

Usability can be defined as the degree to which a given piece of software assists the person sitting at the keyboard to accomplish a task, as opposed to becoming an additional impediment to such accomplishment. The broad goal of usable systems is often assessed using several criteria [Nielsen93]:

- Ease of learning.
- Speed of task completion.
- Error rate.
- Subjective user satisfaction.

So, the performance measures considered for data collection in XRoasters tests were:

Performance Measures:

- The number of completed tasks.
- A count of the number of attempts made before completing the task correctly.
- The run time of the tasks.

4 Discussion and Results

In this chapter the tests analysis and results appear. Three types of results are present: the results obtained from the usability test; those obtained from the subjective user satisfaction survey; and the icon evaluation test results.

4.1 Data Collection

In the data collection three types of measures were considered: the objective ones (performance), the subjective ones (preference), and the subjunctives icons (icons means). The objectives are related to the three performance measures mentioned in last section 3.7.7. The thesis author, while applying the test to the users, collected the data relating to them. Preferences were measured by asking questions as to whether the user agreed or not with some assertions: easy of use, utility of the application, user satisfaction. The icon evaluation was done with a questionnaire, where the user can write the icon meaning that he/she understands, after being given a short introduction to the system.

4.2 Usability Tests

The usability test was developed by strictly following all guidelines established in the test plan, presented in chapter 3.7. Once these tests have been concluded and all the necessary information compiled, a detailed analysis was carried out in order to quantify the exposed variables of evaluation in section 3.7.7.

4.2.1 Icon Evaluation

For the icons evaluation, we take the results from the forms we gave to the users and evaluate them by giving them a mark from 1 to 5, depending on how close they are to the real meaning of the icon, where 5 is Very Good and 1 Very Bad. The results are displayed in Table 2 below.

Table 7 Results Icon Evaluation

USER	ICONS									
	1	2	3	4	5	6	7	8	9	10
User1	4	4	5	5	5	4	5	3	5	5
User2	5	4	5	5	5	5	5	3	5	5
User3	5	5	5	5	5	5	5	5	5	5
User4	5	5	5	5	5	3	3	5	5	5
User5	3	3	5	5	2	5	5	3	5	5
User6	5	5	5	5	5	5	5	5	5	5
User7	5	5	5	5	5	5	5	1	5	5
User8	4	4	5	2	4	2	4	4	5	5
User9	5	5	5	5	5	5	5	5	5	5
User10	5	5	4	4	4	3	3	3	5	5
User11	5	5	3	4	4	5	5	4	5	5
User12	5	5	5	5	5	5	5	5	5	5
User13	5	5	5	5	1	5	5	5	5	5
User14	5	5	5	5	5	4	5	5	5	5
User15	5	5	5	5	5	5	5	4	5	5
User16	5	5	5	5	1	5	5	5	5	5

Descriptive Statistics										
Mean	4.75	4.69	4.81	4.69	4.13	4.44	4.69	4.06	5.00	5.00
Median	5.00	5.00	5.00	5.00	5.00	5.00	5.00	4.50	5.00	5.00
Mode	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
St Dev	0.58	0.60	0.54	0.79	1.45	0.96	0.70	1.18	0.00	0.00
Max	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00	5.00
Min	3.00	3.00	3.00	2.00	1.00	2.00	3.00	1.00	5.00	5.00

5 – Very Good, 4 – Good, 3 – Regular, 2 – Bad 1 – Very Bad.

We carried out this evaluation for all the icons and the results showed that:

- All the users can understand the real icon objectives;
- The vast majority of users scored 5 for all the icons, with a small minority below this value.

On average, the subjects scored 4.7 which is very good.

4.2.2 Objective Measures

For the performance evaluation we took the results from the forms we had given to the users. There are a total of 12 tasks, presented in chapter 3.7.5.2 where you can also see the tasks' descriptions; the analysis was undertaken using these results which are shown below.

4.2.2.1 The number of completed tasks. In table 3 we present the obtained results related to the number to completed tasks.

Table 8 The number of completed tasks

USER	TASK											
	1	2	3	4	5	6	7	8	9	10	11	12
User1	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User2	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User3	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User4	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User5	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User6	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User7	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User8	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes
User9	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
User10	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes

Table 8 shows that all the users can perform the tasks, with just one user having difficulty with task 11. We can therefore consider that the tasks were fully completed.

4.2.2.2 Count of the number of attempts to complete the tasks correctly. In table 4 we present the obtained results relating to the number of attempts to complete the tasks correctly and their descriptive statistics.

Table 9 Attempts to make the task correctly

USER	ATTEMPTS											
	TASKS											
	1	2	3	4	5	6	7	8	9	10	11	12
User1	1	1	1	1	1	1	1	1	1	1	1	1
User2	1	1	2	2	1	1	1	1	3	1	2	1
User3	1	1	1	1	1	1	1	1	1	1	1	1
User4	1	1	1	1	1	1	1	1	1	1	1	1
User5	1	1	1	1	1	1	1	1	1	1	1	1
User6	1	1	1	1	1	1	1	1	1	1	1	1
User7	1	1	1	1	3	1	1	1	2	1	3	1
User8	1	2	1	2	2	1	1	2	1	1	3	1
User9	1	1	1	1	1	1	1	1	1	1	1	1
User10	1	2	1	1	1	1	1	1	1	1	1	1

Descriptive Statistics												
Mean	1	1.2	1.1	1.2	1.3	1	1	1.1	1.3	1	1.5	1
Median	1	1	1	1	1	1	1	1	1	1	1	1
Mode	1	1	1	1	1	1	1	1	1	1	1	1
St Dev.	0	0.42	0.32	0.42	0.67	0	0	0.32	0.67	0	0.85	0
Max	1	2	2	2	3	1	1	2	3	1	3	1
Min	1	1	1	1	1	1	1	1	1	1	1	1

Table 9 shows that in general the tasks were executed in just one attempt; only task 11 presented any difficulty, with the users averaging 1.5 attempts to complete the task. The maximum number of attempts for tasks 5, 9 and 11 were 3. Also for tasks 2, 3, 4 and 8 the maximum number of attempts was 2, the rest of the attempted tasks were completed in 1 attempt.

4.2.2.3 Run time of the tasks. In table 10 we present the results relating to the run time of tasks. The units used for time are seconds.

In table 10 we can observe the time that each user took to complete each task. For task 1, the average time was 25 seconds, and for task 3 it was 16 seconds. These were the two minimum times taken by users to complete a task. The maximum times occurred with tasks 9 and 11, which averaged 112 and 122 seconds respectively.

Table 10 Run time of the tasks

USER	TASKS											
	1	2	3	4	5	6	7	8	9	10	11	12
	<i>Time Seconds</i>											
User1	26	108	23	84	116	46	33	36	90	70	128	87
User2	25	93	41	84	116	34	49	87	247	65	197	65
User3	27	151	8	58	85	20	28	42	128	66	62	73
User4	28	134	5	140	101	32	50	97	96	106	139	104
User5	30	92	7	50	89	27	34	10	107	51	76	99
User6	10	80	10	68	90	32	23	68	101	50	47	157
User7	26	108	23	84	116	46	33	36	90	70	128	87
User8	46	142	10	102	147	37	38	106	123	67	331	72
User9	17	73	10	38	63	17	63	26	55	40	42	52
User10	18	84	18	152	106	13	35	71	81	49	72	91

<i>Descriptive Statistics</i>												
Mean	25	107	16	86	103	30	39	58	112	63	122	89
Median	26	101	10	84	104	32	35	55	99	66	102	87
Mode	26	108	10	84	116	46	33	36	90	70	128	87
St Dev.	9.5	27	11	37	23	11	12	32	52	18	88	29
Max	46	151	41	152	147	46	63	106	247	106	331	157
Min	10	73	5	38	63	13	23	10	55	40	42	52

In the next table we are going to present the total time required for each user to perform all the tasks.

Table 11 Total Run time of the tasks

USER	TOTAL TIME/ MIN.SECONDS
User1	14.22
User2	18.45
User3	12.56
User4	17.3
User5	11.31
User6	12.36
User7	14.22
User8	20.42
User9	8.32
User10	13.25
Descriptive Statistics	
Mean	14.24
Median	13.43
Mode	14.22
St Dev.	3.59
Max	20.42
Min	8.32

In table 11 we can observe the total time required by each user to perform all 12 tasks. With the descriptive statistics we can determine that the average total time for the tasks was 14 minutes and 24 seconds. The most common time taken was 14 minutes and 13 seconds. Almost 50% of the subjects took over 13 minutes and 44 seconds, rest are below this value. The maximum time required was 20 minutes and 25 seconds, and minimum time was 8 minutes and 19 seconds, also the tasks were completed with a median time of 13.43 minutes, with on average, 3.5 units of scale.

In the following figure 11 we can observe the total time behavior.

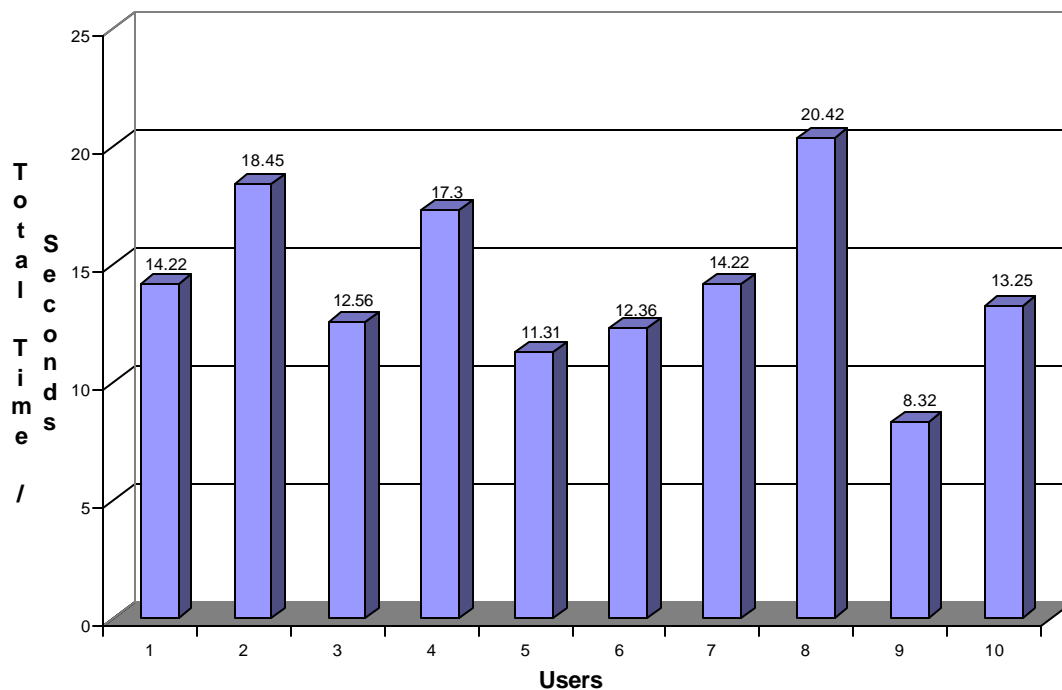


Figure 11 Total Run Time of the tasks

4.2.3 Preference Measures

The preference measures were calculated using questions where the user agreed or not with some assertions: easy of use, utility of the application, user satisfaction. The qualification of the assertions, corresponds to the value given by the user where it indicates how much it was in agreement with the phrase. The scale was from 1 to 7, where 1 is ***Totally Disagree*** and 7 is ***Totally Agree***.

Now we are going to present each assertions with their own evaluation from the user, and also their descriptive statistics.

Assertion 1. Summarily, it found XRoaster easy to use.

Table 12 Assertion 1 Score and Descriptive Statistics

ASSERTION 1	
USER	SCORE
User1	7
User2	7
User3	6
User4	7
User5	6
User6	6
User7	6
User8	6
User9	6
User10	5
Descriptive Statistics	
Mean	6.2
Median	6.0
Mode	6.0
St Dev	0.6
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 12, we can observe that the users are *Strongly in Agreement* with assertion 1. The most common score was 6. Almost all the users are over 6.0, the rest are below this value. On average the users rate assertion 1 with 6.2, with a standard deviation of 0.6. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the high values, so the users *Strongly Agree* that XRoaster is easy to use.

Assertion 2. The terminology was clear and precise.

Table 13 Assertion 2 Score and Descriptive Statistics

ASSERTION 2	
USER	SCORE
User1	7
User2	7
User3	7
User4	6
User5	7
User6	7
User7	6
User8	5
User9	7
User10	5
Descriptive Statistics	
Mean	6.4
Median	7.0
Mode	7.0
St Dev	0.8
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 13, we can observe that the users *Strongly Agree* with assertion 2. The most common score was 7, and on average the users rate assertion 2 with a score of 6.4, with a standard deviation of 0.8. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the higher values, so the users are *Strongly Agree* that the XRoaster terminology was clear and precise.

Assertion 3. Always knew that tasks executed satisfactorily.

Table 14 Assertion 3 Score and Descriptive Statistics

ASSERTION 3	
USER	SCORE
User1	7
User2	6
User3	7
User4	1
User5	7
User6	5
User7	5
User8	5
User9	7
User10	5
Descriptive Statistics	
Mean	5.5
Median	5.5
Mode	7.0
St Dev	1.8
Max	7.0
Min	1.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 14, we can observe that the users *Partially Agree* with assertion 3. The most common score was 7, on average the users rate aseveration 3 with a score of 5.5, with a standard deviation of 1.8. The maximun score was 7.0 and the minimum was 1.0. The scores tend to be located in the average values, so the users *Partially Agree* that they always knew that they had executed the tasks satisfactorily.

Assertion 4. You have the necessary knowledge to make the tasks.

Table 15 Assertion 4 Score and Descriptive Statistics

ASSERTION 4	
USER	SCORE
User1	7
User2	6
User3	7
User4	5
User5	6
User6	6
User7	7
User8	6
User9	7
User10	7
Descriptive Statistics	
Mean	6.4
Median	6.5
Mode	7.0
St Dev	0.7
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 15, we can observe that the users *Strongly Agree* with assertion 4. The score that was most common was 7. On average the users rate assertion 4 with a score of 6.4, with a standard deviation of 0.7. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the high values, so the users *Strongly Agree* that he/she has the knowledge necessary to do the tasks.

Assertion 5. You felt comfortable using the interface.

Table 16 Assertion 5 Score and Descriptive Statistics

ASSERTION 5	
USER	SCORE
User1	6
User2	7
User3	7
User4	6
User5	7
User6	7
User7	6
User8	6
User9	7
User10	6
Descriptive Statistics	
Mean	6.5
Median	6.5
Mode	6.0
St Dev	0.5
Max	7.0
Min	6.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 16, we can observe that the users *Strongly Agree* with assertion 6. The most common score was 6.0. On average the users rate assertion 6 with a score of 6.5, given a standard deviation of 0.5 units of scale. The maximum score was 7.0 and the minimum was 6.0. The scores tend to be located in the high values, so the users *Strongly Agree* that he/she felt comfortable using the interface.

Assertion 6. Do you think that an appropriate used of the color in the screen.

Table 17 Assertion 6 Score and Descriptive Statistics

ASSERTION 6	
USER	SCORE
User1	7
User2	7
User3	6
User4	7
User5	4
User6	7
User7	7
User8	6
User9	7
User10	5
Descriptive Statistics	
Mean	6.3
Median	7.0
Mode	7.0
St Dev	1.1
Max	7.0
Min	4.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 17, we can observe that the users *Strongly Agree* with assertion 6. The most common score was 7.0. The average score the users gave assertion 6 was 6.3, with a standard deviation of 1.1 units of scale. The maximum score was 7.0 and the minimum was 4.0. The scores tend to be located in the high values, so the users are *Strongly Agree* that an appropriate color was used on the screen.

Assertion 7. There were not problems with navigation in the interface.

Table 18 Assertion 7 Score and Descriptive Statistics

ASSERTION 7	
USER	SCORE
User1	7
User2	6
User3	4
User4	6
User5	7
User6	6
User7	5
User8	6
User9	7
User10	6
Descriptive Statistics	
Mean	6.0
Median	6.0
Mode	6.0
St Dev	0.9
Max	7.0
Min	4.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 18, we can observe that the users *Strongly Agree* with assertion 7. The most common score was 6.0. On average the users rate assertion 7 with a score of 6.0, given a standard deviation of 0.9 units of scale. The maximum score was 7.0 and the minimum was 4.0. The scores tend to be located in the higher values, so the users *Strongly Agree* that there were not problems with navigation in the interface.

Assertion 8. Do you think that there was not too much information in the screen.

Table 19 Assertion 8 Score and Descriptive Statistics

ASSERTION 8	
USER	SCORE
User1	7
User2	6
User3	6
User4	5
User5	6
User6	4
User7	4
User8	6
User9	7
User10	5
Descriptive Statistics	
Mean	5.6
Median	6.0
Mode	6.0
St Dev	1.1
Max	7.0
Min	4.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 19, we can observe that the users *Partially Agree* with assertion 8. The score that more repeated was 6.0. On average the users rate assertion 8 at 5.6, with a standard deviation of 1.1 units of scale. The maximum score was 7.0 and the minimum was 4.0. The scores tend to be located in the higher values, so the users *Partially Agree* that there was not too much information in the screen.

Assertion 9. Do you think that there was a suitable use of the space in the screen.

Table 20 Assertion 9 Score and Descriptive Statistics

ASSERTION 9	
USER	SCORE
User1	7
User2	7
User3	7
User4	6
User5	5
User6	7
User7	6
User8	5
User9	7
User10	5
Descriptive Statistics	
Mean	6.2
Median	6.5
Mode	7.0
St Dev	0.9
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 20, we can observe that the users *Strongly Agree* with assertion 9. The most common score was 7.0. On average the users rate assertion 9 with a score of 6.2, with a standard deviation of 0.9 units of scale. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the high values, so the users *Strongly Agree* that there was a suitable use of space on the screen.

Assertion 10. The description of tasks was correct.

Table 21 Assertion 10 Score and Descriptive Statistics

ASSERTION 10	
USER	SCORE
User1	6
User2	7
User3	7
User4	7
User5	7
User6	7
User7	6
User8	6
User9	6
User10	7
Descriptive Statistics	
Mean	6.6
Median	7.0
Mode	7.0
St Dev	0.5
Max	7.0
Min	6.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4 Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 21, we can observe that the users *Strongly Agree* with assertion 10. The most common score was 7.0. On average the users rate assertion 10 at 6.6, with a standard deviation of 0.5. The maximum score was 7.0 and the minimum was 6.0. The scores tend to be located in the high values, so the users *Strongly Agreement* that the description of tasks was correct.

Assertion 11. Highly Technical.

Table 22 Assertion 11 Score and Descriptive Statistics

ASSERTION 11	
USER	SCORE
User1	7
User2	6
User3	6
User4	6
User5	5
User6	6
User7	5
User8	3
User9	1
User10	6
Descriptive Statistics	
Mean	5.1
Median	6.0
Mode	6.0
St Dev	1.8
Max	7.0
Min	1.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 22, we can observe that the users *Partially Agree* with assertion 11. The score that was most repeated was 6.0. On average the users rate assertion 11 at 5.1, with a standard deviation of 1.8 units of scale. The maximum score was 7.0 and the minimum was 1.0. The scores tend to be located in the average values, so the users *Partially Agree* that XRoaster was Highly Technical.

Assertion 12. Reliable.**Table 23 Assertion 12 Score and Descriptive Statistics**

ASSERTION 12	
USER	SCORE
User1	7
User2	6
User3	6
User4	6
User5	6
User6	7
User7	6
User8	5
User9	6
User10	5
Descriptive Statistics	
Mean	6.0
Median	6.0
Mode	6.0
St Dev	0.7
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 23, we can observe that the users *Strongly Agree* with assertion 12. The most common score given was 6.0. On average the users rate assertion 12 with a score of 6.0, given a standard deviation of 0.7. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the high values, so the users *Strongly Agree* that XRoaster was reliable.

Assertion 13. Easy To use .

Table 24 Assertion 13 Score and Descriptive Statistics

ASSERTION 13	
USER	SCORE
User1	7
User2	7
User3	6
User4	7
User5	6
User6	6
User7	5
User8	5
User9	7
User10	6
Descriptive Statistics	
Mean	6.2
Median	6.0
Mode	7.0
St Dev	0.8
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 24, we can observe that the users *Strongly Agree* with assertion 13. The most common score was 7.0. On average the users rate assertion 13 at 6.2, with a standard deviation of 0.8. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the high values, so the users *Strongly Agree* that XRoaster was easy to use.

Assertion 14. Useful.**Table 25 Assertion 14 Score and Descriptive Statistics**

ASSERTION 14	
USER	SCORE
User1	7
User2	7
User3	7
User4	7
User5	7
User6	7
User7	7
User8	6
User9	7
User10	6
Descriptive Statistics	
Mean	6.8
Median	7.0
Mode	7.0
St Dev	0.4
Max	7.0
Min	6.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 25, we can observe that the users *Totally Agree* with assertion 14. The score that was most repeated was 7.0. On average the users rate assertion 14 with a score of 6.8, given a standard deviation of 0.4 units of scale. The maximum score was 7.0 and the minimum was 6.0. The scores tend to be located in the high values, so the users *Totally Agree* that XRoaster was useful.

Assertion 15. Friendly.**Table 26 Assertion 15 Score and Descriptive Statistics**

ASSERTION 15	
USER	SCORE
User1	6
User2	7
User3	6
User4	6
User5	6
User6	6
User7	5
User8	6
User9	7
User10	5
Descriptive Statistics	
Mean	6.0
Median	6.0
Mode	6.0
St Dev	0.7
Max	7.0
Min	5.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 26, we can observe that the users *Strongly Agree* with assertion 15. The most common score was 6.0. On average the users rate assertion 15 with a score of 6.0, given a standard deviation of 0.7. The maximum score was 7.0 and the minimum was 5.0. The scores tend to be located in the high values, so the users *Strongly Agree* that XRoaster are friendly.

Assertion 16. I Like the XRoaster tool.

Table 27 Assertion 16 Score and Descriptive Statistics

ASSERTION 16	
USER	SCORE
User1	7
User2	7
User3	6
User4	7
User5	7
User6	6
User7	7
User8	6
User9	7
User10	6
Descriptive Statistics	
Mean	6.6
Median	7.0
Mode	7.0
St Dev	0.5
Max	7.0
Min	6.0

Explanation: 7. Totally Agree, 6. Strongly Agree, 5. Partially Agree, 4. Neutral, 3. Partially Disagree, 2. Strongly Disagree, 1. Totally Disagree.

In table 27, we can observe that the users *Totally Agree* with assertion 16. The score that most repeated was 7.0. On average the users rate assertion 16 with a score of 6.6, given a standard deviation of 0.5 units of scale. The maximum score was 7.0 and the minimum was 6.0. The scores tend to be located in the high values, so the users *Totally Agree* that he/she likes XRoaster Tool.

Finally, I am going to present the total rate in user satisfaction.

In general we can conclude that there is a trend indicating that the XRoaster's interface is very friendly for the user, because we found by means of the statistical analysis that the degree of satisfaction from the user is acceptable. The Figure 12 depicts the rate in user satisfaction. On average the users rate XRoaster with a score of 6.1. The maximum score was 6.8 and the minimum was 5.4. The scores tend to be located in the higher values, so the users have a good satisfaction working with XRoaster. Clearly, a more comprehensive study to provide a stronger case for XRoaster, but our evidence show the trend for success and usability of XRoaster.

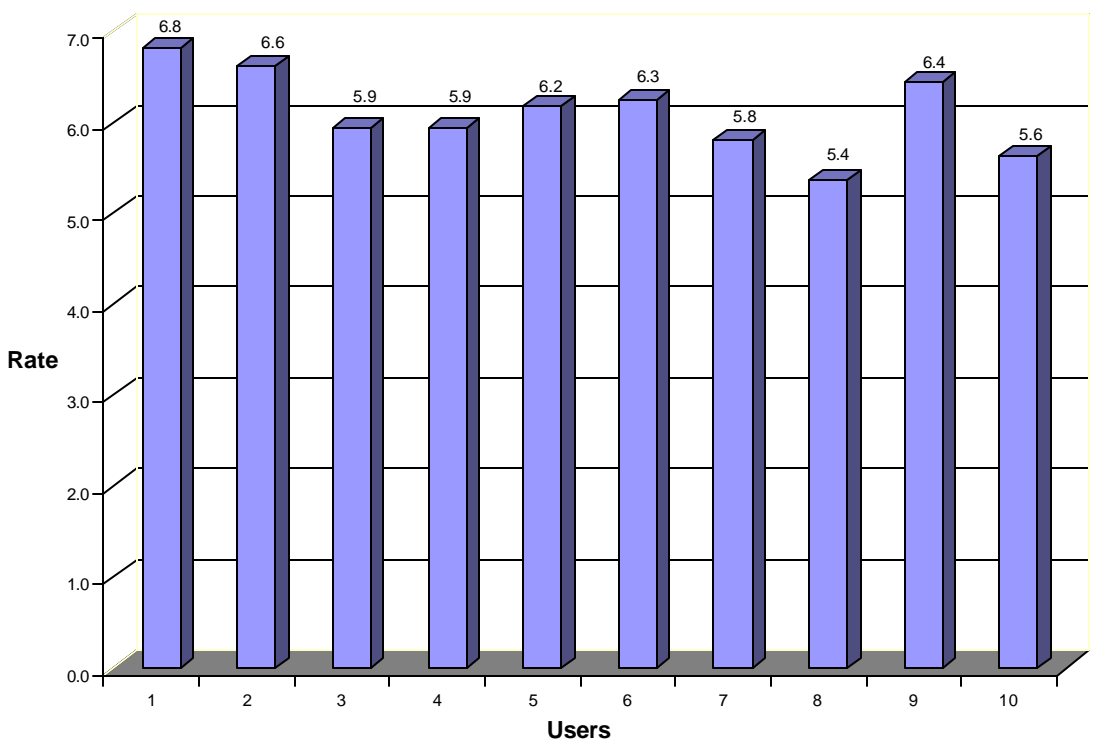


Figure 12 User Satisfaction

5 Conclusions and Future Work

In this thesis we identified a problem that is schema mapping in Database Middleware Systems. We introduced XRoaster, an XML-based tool for distributed schema mapping that is fully integrated with the catalog and execution engine of the Middleware System. This application easily creates the global schema and the infrastructure needed to register data sources, user defined code and schema mappings in Database Middleware Systems.

5.1 Conclusions of Research

XRoaster makes data sharing among the data sources possible. The framework discussed here provides a generic Databases Middleware Systems Architecture for integration of enterprise-wide applications or data. The benefits of XRoaster are: 1) Tool integration capability; 2) extensibility of the information system, allowing for the addition of new local data sources; 3) exchange of data formats among data sources; and 4) an integrated data view that allows sharing data between distributed applications. We developed a Visual tool with a GUI, which has the options to do the schema mapping process. The goal is to handle all schema management tasks from this tool. XRoaster minimizes the amount of code that the administrators need to write in order to perform schema mapping operations. XRoaster maps the data from the local schema into the global schema and also implements a schema mapping framework. The GUI helps the administrator to create the schema mapping rules and the XML metadata documents that contain the types of query capabilities of the local and global source.

There are several methods of evaluating the usability of a tool or system: Heuristic analysis, and usability tests with users. In this thesis an usability evaluation appears using the method of usability test with users, that allows us to identify problems in an XRoaster interfaces.

We collected three types of measures: Objective (performance), the subjective ones (Preference), and subjunctives icons (icons means).

The objectives parameters were related to three performance measures: 1) The number of completed tasks, 2) A count of the number of attempts to complete the task correctly, and 3) Run time of the tasks. The preference measures were made by means questions where the user agreed or not with some assertions: ease of use, utility of the application, and user satisfaction. The icon evaluation was done with a questionnaire, where the user can write the icon meaning that he/she understands, after being given a short introduction to the system. With the results from the user satisfaction tests we can ensure that the user had a very good user satisfaction with XRoaster, and with the results from icon evaluation we can affirm that the XRoaster's icons convey the correct meaning to the user, and that he/she can relate the icon with the action that is executed.

The usability tests were completed, making it is possible to determine the XRoaster menus capacity, to establish their functionality, and identify navigation problems. In general terms it is possible to conclude that there is a clear trend that supports the claim that the XRoaster interface is very user friendly, since the statistical analysis showed that the satisfaction degree for the user is acceptable. In our study, 100% of the tasks were carried out satisfactorily.

Based on the results of the usability test, the following trends in usability qualities can be stated and are classified according to the theory of Jacob Nielsen:

Natural Dialogue: The XRoaster's interface is clear, because the user can successfully perform all tasks that he/she has to do. For example the dialogue boxes are in the user's native language.

Consistency: In general there were no problems regarding interface inconsistency.

Feedback: For some tasks, the system does not have a way of displaying to the user that he/she has successfully completed a task. So the XRoaster's feedback needs to be improved.

Clearly Marked Exits: The user has an easy way out from several situations that he/she can find.

Shortcuts: The application can be operated with the knowledge of just a few general rules, and also the user can perform frequently used operations fast, using dialogue shortcuts.

5.2 Future Work

The main limitation of this study was the small sample that we had available to perform the usability tests. Hence, as part of the future work in the system it is necessary to make larger usability study. One strategy is to make contacts in other universities in order to find expert users to perform the test. This will enable us to make a stronger case that the XRoaster is indeed a very usable and efficient tool.

In addition, we want to improve all the processes of Catalog Management and Schema Mapping. The GUI's will help an administrator to create a schema mapping and also their profiles, we want that the end user will follow a "wizard-based" approach where XRoaster will suggest possible mappings, and configuration options to the user, these suggestions will be based on local information. Also, XRoaster should be able to give warnings to the user indicating possible conflicting mappings or other suspicious operations.

The next step is to extend the data model of the catalog to capture more characteristics like data source properties, federation topology and their properties. Also we want the application to be even more user friendly, and for that we need to develop a tool for on-line help that describes how XRoaster works.

Bibliography

[Bertino01] Bertino Elisa and Catania Barbara, Integrating XML and Databases, *IEEE Internet Computing*, Volume: 5 Issue: 4, Page(s): 84 -88, 2001.

[Bernstein01] Bernstein Philip, Bergstraesser Thomas. MetaData Support for Data Transformations Using Microsoft Repository. *IEEE Data Engineering Bulletin 22* Page(s): 9-14. 1999

[Carrer01] Carrer Marco, Jhosi Ashok, Lin Paul, Srivastava. Metadata Transformation and Management with Oracle Intermedia. *IEEE Data Engineering Bulletin. 22(1)* Page(s) 15-18. 1999.

[Claypool01] Claypool Kajal, Rundensteiner Elke. Flexible Database Transformations: The Serf approach. *IEEE Data Engineering Bulletin*, 1999.

[Cho97] Cho Haengrae . Catalog Management in Heterogeneous Distributed Database Systems. *Communications, Computers and Signal Processing, 1997. '10 Years PACRIM 1987-1997 - Networking the Pacific Rim'*. Victoria – Canada, 1997.

[Hass01] Hass L. M. Transforming Heterogeneous Data with Database Middleware: Beyond Integration. *IEEE Internet Computing*. 2001.

[Larson00] Larson James A. and Larson Carol L., "Migrating Data to an Integrated Database" *Enterprise Systems Integration*, Page(s): 401-409, Boca Raton, Florida, 2000.

[Ling01] Ling Yang, Renée Miller, Laura Hass, Ronald Fagin. Data-Driven Understanding and Refinement of Schema Mappings. *ACM SIGMOD 2001*, Santa Barbara California, USA. 2001

[Lakshmanan02] Lakshmanan Laks, Sadri Fereidoon, Subramanian Iyer. SchemaSQL – A Language for interoperability in Relational Multi-database systems. *In Proceedings of 22nd International Conference on Very Large Data Bases*, Bombay, India. 1996.

[Mische00] Mische Michael A., "Defining Systems Integration" *Enterprise Systems Integration*, Page(s): 3-10, Boca Raton, Florida, 2000.

[Miller2000] Miller Reneé, Hass Laura, Hernandez Mauricio. Schema Mapping as Query Discovery. *International conference on very Large Data Bases (VLDB)*, Morgan Kaufmann, Page(s): 77-88. 2000.

[Molina03a] Molina Ana Mercedes, Rodriguez-Martinez Manuel. XRoaster: A Tool for Catalog Management on Databases Middleware Systems. *7th IASTED International Conference on: Internet and Multimedia Systems and Applications - IMSA*, Honolulu-Hawaii, USA, 2003.

[Molina03b] Molina Ana Mercedes, Rodriguez-Martinez Manuel. XRoaster An Xml-Based Visual Tool For Data Sharing And Integration. *The Second IASTED International Conference on: Information and Knowledge Sharing - IKS*, Scottsdale-Arizona, USA, 2003.

[Nielsen93] Nielsen Jakob. *Usability Engineering*. Morgan Kaufmann Publishing Company, San Francisco, 1993.

[O'Brien99] O'Brien, James A. *Management Information Systems*. McGraw-Hill International Editions, USA, 1999.

[Orenstein99] Orenstein Jack. Supporting Retrievals and Updates in an Object/Relational Mapping Systems. *IEEE Data Engineering Bulletin* 22(1) Page(s): 50-54, 1999.

[Posch00] Posch Raymond J., "Using Middleware for Interoperable Systems" *Enterprise Systems Integration*, Page(s): 185-196, Boca Raton, Florida, 2000.

[Ramakrishnan03] Ramakrishnan Gehrke. *Database Management Systems*. Third Edition. 2003.

[Rodríguez99a] Rodríguez Manuel, Roussopoulos Nick. MOCHA: A Self-Extensible Database Middleware System for Distributed Data Sources. *Technical Report UMIACS-TR 200005, CS-TR 4105*, University of Maryland, 2000.

[Rodríguez00b] Rodríguez Manuel, Roussopoulos Nick. Automatic Deployment of Application-Specific Metadata and Code in MOCHA. *In Proc. 7th EDBT Conference*, Konstanz, Germany, March 2000.

[Shaosong02] Shaosong Xu, Hoh In, and Martin S. Feather. A Shared Information System Architecture for Integrating Risk Management Tools: A Case Study . *26th Annual International Computer Software and Applications Conference*. Oxford, England, 2002.

Annex A Letter Introduction to the User

This annex has the letter that is given to the user at the beginning of the test.

Mayagüez, 15 de Octubre de 2003

Buenos días (tardes):

Mi nombre es: **Ana Mercedes Molina** y trabajaré con Usted en la sesión del día de hoy, explicándole el objetivo de este estudio.

La razón por la que usted se encuentra aquí es porque necesitamos probar la aplicación XRoaster para determinar la funcionalidad y para ello requerimos su ayuda y colaboración; estas pruebas se hacen únicamente con carácter didáctico, de manera que se mantendrá su identidad en reserva.

Las tareas a trabajar son las típicas aprendidas que se efectuaran en XRoaster. Se requiere que usted las haga con la velocidad, nivel de atención y detalle necesarios que usted normalmente realiza. No trate de esforzarse por hacer su mejor desempeño, pues no se examinarán sus destrezas y conocimientos, sino el **Prototipo** y este puede que no trabaje como usted quisiera. Puede hacer preguntas en cualquier momento, pero es posible que no se le puedan contestar ya que este es un estudio del sistema y de sus métodos de ayuda, por esto es necesario ver cómo trabaja usted de manera independiente.

Durante la sesión de hoy, también se solicitara que conteste un formulario, en el cual es **MUY** importante su **veracidad** ya que el propósito de la prueba es descubrir las ventajas y desventajas de esta aplicación.

Mientras usted trabaja, se estarán tomando algunas notas y sincronizaciones para evaluar resultados finales del sistema.

En caso que sienta que no puede llevar a cabo una tarea, esta en la libertad de continuar con la siguiente.

Si tiene alguna pregunta, puede hacerla, y seguidamente se dará inicio a la prueba.

Annex B Forms Data Collection

This annex presents the form used to collect a number of variables to be evaluated at the user test.

FORMULARIO DE RECOLECCION DE DATOS XROASTER

Identificación de Usuario: _____

Fecha: _____


Hora Inicial: _____

Hora Final: _____

TAREA	NÚMERO DE INTENTOS	COMPLETADA		TIEMPO
		SI	NO	
INGRESAR A XRoaster				
VISUALIZAR ESQUEMA LOCAL				
CONECTARSE A ESQUEMA GLOBAL				
VISUALIZAR ESQUEMA GLOBAL				
ADICIONAR NUEVO AUTOR y DAP				
LLENAR CATALOGO				
INGRESAR METADATA PARA TABLA LOCAL				
INGRESAR METADATA PARA TABLA GLOBAL				
CREAR TIPO DE DATO y FUNCION				
CREAR UNA REGLA DE MAPEO DIRECTA Y DE EXPRESION				
CREAR UNA REGLA DE MAPEO MEDIANTE EL USO DE FUNCIONES				
VISUALIZAR MAPPING RULES				
Subtotal				
Total				

Annex C Presentation of XRoaster


For the users group, a small presentation of the XRoaster tool was made as an introduction or pre-training in order to explain the XRoaster functionality.



XRoaster

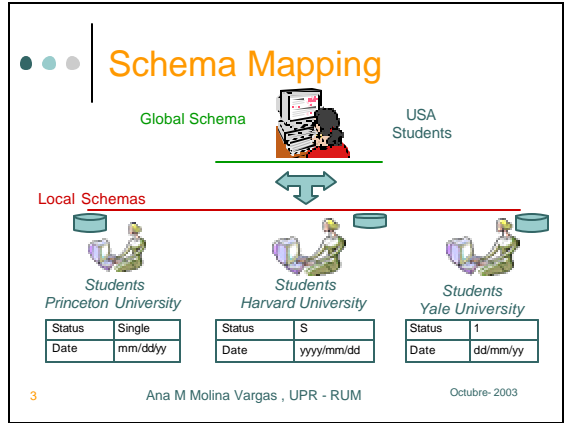
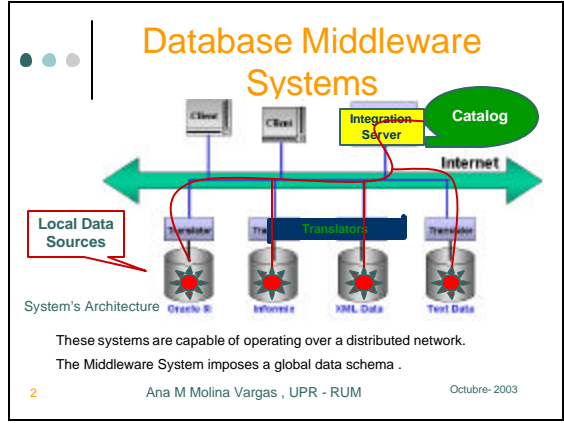
A Tool for Catalog Management
on Middleware Database Systems

Ana Mercedes Molina Vargas
ana.mercedes@ece.uprm.edu
Graduate Student



Manuel Rodriguez, Ph.D.
manuelr@acm.org
Research Advisor

XRoaster UPR - RUM
University of Puerto Rico,
Mayaguez Campus -
Department of Electrical and
Computer Engineering.



Annex D Instructions Test

In the following annex the instructions to the users are explained. This contains a description of the tasks to undertake, each one of which is presented on a separate page.

INSTRUCCIONES

Siga una a una las instrucciones que se le dan a continuación:

TAREA 1.

Conectarse a la Base de Datos “*Pubs*”, como conexión de esquema local, donde:

User : sa

Password:

Data Base Space: “*Pubs*”

TAREA 2.

Visualizar el esquema local y llenar los datos que se solicitan en la hoja de resultados con la tabla local “*publishers*”.

TAREA 3.

Hacer conexión global con la Base de Datos “*NorthWind*”.

TAREA 4.

Visualizar el esquema global y llenar los datos que se solicitan en la hoja de resultados con la tabla global “*Products*”.

TAREA 5.

Adicionar los datos de un “*Author*” y “*Data Access Provider*” para identificar la nueva fuente de datos que se va a adicionar al esquema global.

Información de Nuevo “*Author*”:

ID Autor: 11

Last Name: Perez
First name: Pepito
Email: pepito@hotmail.com

Información de Nuevo “*Data Access Provider*”

ID DAP: 123
Host Name: CRRLAPTOP01
IP Address: 160.160.255.255
User: Juanito
Password: Flor

TAREA 6.

Ejecutar la opción de poblar (“populate”) el catalogo de XRoaster, verificando que el esquema Global es “*NorthWind*” y el esquema local es “*Pubs*”, además verifique que el id de “*Author*” y id de “*Data Access Provider*” son los que creo en la tarea numero 5.

TAREA 7.

Ingresar metadata en las tablas locales del sistema, con la siguiente información:

Para la Tabla Local “*publishers*”

Columna: “*Pub_id*”
Size: 200
Min Value: 0
Max Value: 2000

Columna “*city*”
Size: 400
Min Value: 2000
Max Value: 4000

Columna “*state*”
Size: 600
Min Value: 4000
Max Value: 6000

TAREA 8.

Ingresar metadata en las tablas globales del sistema, con la siguiente información:

Tabla Global: “*EmployeeTerritories*”

Columna: “*EmployeeID*”

Size: 500

Min Value: 0

Max Value: 2000

Columna “*TerritoryID*”

Size: 500

Min Value: 0

Max Value: 4000

TAREA 9.

Ingresar un nuevo tipo de dato (“*Type*”) y operación (“*Operator*”).

Información del Nuevo “*Type*”

Id: 1

Auhorid: Identificación de usuario que creó al comienzo del sistema.

Name: MIQuadPoly

Size: 33

Información del Nuevo “*Operator*”

Id. 11

Authorid: Identificación de usuario que creó al comienzo del sistema.

Arguments: 2

Name: overlaps

TAREA 10.

Crear las reglas que permiten hacer el “*schema mapping*”.

Hacer un Schema Mapping Rule entre las tablas:

Global: “*Employees*”

Local: “*Authors*”

Schema Mapping tipo “*Direct*” entre las columnas “*Employeeid*” y la columna “*au_id*”.

Schema Mapping tipo “*Expression*” para la columna “*PostalCode*” con un valor de 00681.

TAREA 11.

Crear regla de mapeo pero con las opciones de funciones.

Hacer un Schema Mapping Rule entre las tablas:

Global: “*Employees*”

Local: “*Authors*”

Crear un Schema Mapping Rule Tipo “*Function*” entre las columnas:

Tabla Global “*Employees*”

Columna “*LastName*”

Tabla Local “*Authors*”

Columnas: “*au_lname*”

“*au_fname*”

Aplicando la función “*CONCAT*” de tipo string.

TAREA 12.

Visualizar la metadata de los “*schema mapping rules*” creados y llenar los datos que se solicitan en la hoja anexa.

Annex E Results Form

The following results form sets out for the user answers to possible questions about the executed tasks.

FORMULARIO DE RECOLECCION DE DATOS XROASTER

Identificación de Usuario: _____

Fecha: _____

Tarea 2: Metadata Local Table	
Schema Local	
Nombre Tabla Solicitada	
Metadata:	
Llave primaria	
Tipo de Dato	
Tamaño	
Es Nulo	
Tarea 4: Metadata Global Table	
Schema Global	
Nombre Tabla Solicitada	
Metadata:	
Llave primaria	
Tipo de Dato	
Tamaño	
Es Nulo	
Tarea 12 : Schema Mapping	
Id Mapping	
Direct	
Mapping Expression	
Id Mapping	
Direct	
Mapping Expression	
Id Mapping	
Direct	
Mapping Expression	

Annex F Questionnaire User Satisfaction

This questionnaire compiles subjective information and the perception of the participants.

CUESTIONARIO SATISFACCION DE USUARIO XROASTER

Identificación de Usuario: _____

Marque con una X debajo de la columna que indique cuan de acuerdo esta usted con cada una de las siguientes aseveraciones, donde: la columna significa **1.** Nada de Acuerdo y la columna **7.** significa Totalmente de Acuerdo


	1	2	3	4	5	6	7
En conjunto, encontré XRoaster fácil de usar.							
La terminología fue clara y precisa.							
Siempre sabía que ejecutaba las tareas satisfactoriamente.							
Tenia el conocimiento necesario para realizar las tareas							
Se sintió cómodo utilizando la interfaz.							
Piensa que se hizo un apropiado uso del color en la pantalla.							
No existían problemas con la navegación en la interfaz.							
Piensa que no había demasiada información en la pantalla.							
Piensa que existía una adecuada utilización del espacio en la pantalla.							
La descripción de tareas fue correcta.							
Altamente Técnico							
Confiable							
Fácil de Utilizar							
Util							
Amigable							
Me gusto							

Comentarios Adicionales:

Annex G. Icon Evaluation Form

This form compiles the perception of the participants about the XRoaster icons.

Explique brevemente que entiende por cada uno de los siguientes iconos:

 Local Connection	<hr/> <hr/> <hr/> <hr/>
 Global Connection	<hr/> <hr/> <hr/> <hr/>
 Populate Catalog	<hr/> <hr/> <hr/> <hr/>
 Type	<hr/> <hr/> <hr/> <hr/>
 Operator	<hr/> <hr/> <hr/> <hr/>
 Local Table	<hr/> <hr/> <hr/> <hr/>
 Global Table	<hr/> <hr/> <hr/> <hr/>
 Mapping	<hr/> <hr/> <hr/> <hr/>
 About	<hr/> <hr/> <hr/> <hr/>
 End	<hr/> <hr/> <hr/> <hr/>