P-2740

XML SERVICES - Integration Tool

By

G.S.Sivagurunathan

Register Number: 71206621051

Of

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements
For the award of the degree

Of

MASTER OF COMPUTER APPLICATIONS

ANNA UNIVERSITY
CHENNAI 600 025

July, 2009

BONAFIDE CERTIFICATE

Certified that this project report titled **"XML SERVICES - Integration Tool"** is the bonafide work of **"Mr.G.S.Sivagurunathan"** (Register Number: **71206621051**) who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

N. Jayakanthan
SUPERVISOR

HEAD OF THE DEPARTMENT

Submitted to Project and Viva Examination held on 06.07.09

6/7/09
INTERNAL EXAMINER

6/7/09
EXTERNAL EXAMINER

ABSTRACT

KEYWORDS: Data source and their types, namespace, modeler map, schema view or schema tree, command view or command tree, Biztalk assembly.

XMLService is a software product to enable the integration of Honeywell software systems and applications with ERP (Enterprise Resource Planning) systems. This product also enables integration with supply chain management systems and other 3rd party systems.
The system focuses on the development of a common and generic application framework to build various applications of business groups. This report deals with designing of XML Service which provides generic capabilities to support business-to-manufacturing (B2M) integration needs. XML Service provides generic capabilities by using Standards and Microsoft technologies without relying on any other vendor specific EAI technology. XML Service enables easy integration of Honeywell software systems and applications with ERP systems.

The objectives of the system include having a common look and feel across various Honeywell applications, co-existing with other Honeywell applications in a UI framework, Consume historical data, journal events, generate alarms or notifications destined for the operator, to allow application data to be alarmed and historized, share information with other Honeywell applications etc.

ACKNOWLEDGEMENT

I extend my sincere gratitude to **Dr.R.Annamalai,** Vice Principal, Kumaraguru College of Technology, Coimbatore, for permitting me to undertake this project.

I wish to express my sincere thanks to **Dr. M.Gururajan**, Professor and Head, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for allowing me to do a project at Honeywell Technology Solution Lab, Bangalore and for his continuous support.

I extend my gratitude and great full thanks to my Co-ordinator guide, **Mrs. V.Geetha,** Assistant Professor, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for her continuous support and guidance throughout the project.

I wish to record my gratitude and great full thanks to my faculty guide, **Mr.N.Jayakanthan,** Lecturer, Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for his continuous kindly help, support and guidance throughout the project.

I would like to mention my sincere gratitude to my industry guide, **Mr. Ananth** , Team Leader (ITSS-ACS), and **Mr.Theerthagiri**, Engineer Honeywell Technology Solutions Lab Pvt. Ltd, Bangalore, with their executive expertise and dynamic personalities guided and helped me throughout the course of my project.

Thanks to my beloved parents, friends and everyone else who has been directly or indirectly supportive during the course of this project.

Last but not the least I like to express my sincere gratitude to the God Almighty who showers his blessings throughout this entire period of time.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

HTTP : HyperText Transfer Protocol
XML : Extensible Markup language
ERP : Enterprise Resourse Planning
EAI : Experion Application Intergration
WPKS : Work Process Knowledge System
UML : Unified Modeling Language
W3C XSD : World wide web consortium XML Schema Defination
SQL : Structured Query Language
EAF : Experion Application Framework
FCL : Framework Base Class Library
CLR : Common Language Runtime .
CTS : Common type system
MSIL : Microsoft Intermediate Language
JIT : Just In Time
IL : Intermediate Language
DLL : Dynamic Link Library
DTD : Document Type Definition
DOM : Document Object Model
SAX : Simple API for XML
OLE DB : Object Linking and Embedding, Database
DST : Data Source type
DSA : Data Source access
DCS : Diatributed Control System

CHAPTER 1

INTRODUCTION

1.1 ORGANIZATION OF THE REPORT

This section clearly explains how the report is organized, the associated goals and purpose of each and every chapter discussed for this project "**XML SERVICES**- An Integration Tool".

CHAPTER 1

Chapter 1 gives a brief overview about the project. It starts with the organization profile and explains about the business domains that are related to the Honeywell Technology Solutions Lab (HTSL). The role of languages and technologies that are used for the implementation is followed by the work environment. Finally a brief introduction to the XML SERVICES (Integration Tool) is explained.

CHAPTER 2

Chapter 2 explains the system study that is made for this project. Its starts with the introduction to XML SERVICES Modeler. Then explains the Modeler Basics, Modeler's functionalities and interdependencies of the applications and finally the Modeler's features.

CHAPTER 3

Since the architectural design of this XML SERVICES (Integration Tool) is already done and approved by Honeywell, the study of the architecture that is made for this project is presented in this chapter. The architecture of XML SERVICES is defined first, followed by various levels of diagrams such as Context diagrams, Dataflow

diagrams, and sample use cases, followed by a data dictionary giving description of various terms involved with XML SERVICES.

CHAPTER 4

This chapter is dedicated for explaining various Management features of XML SERVICES MODELER. It is followed by explaining the process involved in the Lookup builder. Finally the Modeler configuration is explained.

CHAPTER 5

This chapter is gives the components of XML SERVICES Configuration Studio namely Modeler, Message Configuration and Tools.

CHAPTER 6

This chapter describes the testing phase of the modules which are implemented. It starts with explaining different kinds of testing followed by sample test case report which is done for the implemented modules.

CHAPTER 7

This chapter gives a conclusion about the project.

1.2 ORGANIZATION PROFILE

Honeywell International was founded in 1885 by Albert Butz. It is a USD 33 billion diversified technology and manufacturing leader, serving customers worldwide with,

- Aerospace products and Services
- Automation and Control Solutions
- Automotive products
- Specialty materials

Honeywell Technology Solutions (HTS) is an integral arm of Honeywell International. It began in October 1994 in Bangalore by Dr. Krishna Mikkilineni. Honeywell Technology Solutions Lab (HTSL), a division of Honeywell International, provides an array of services for aerospace and military customers. Activities include technical and engineering services, space system services, logistics and sustainment, program management services, and IT and communication services. NASA is HTSL's largest customer. We provide value, expertise, competitive pricing and the highest quality standards available. Our customers can expect proven leadership processes, technical excellence and cost management. We offer more than just connections, we offer solutions.

The four Major Business Units of HTSL are:

- Aerospace products and Services
- Automation and Control Solutions
- Automotive products
- Specialty materials

ABOUT GROUP:

IT SERVICES & SOLUTIONS (ITSS):

ITSS is a business unit of HTS, partners with various Honeywell businesses across the globe. The key driver for ITSS is value maximization for Honeywell businesses through application of information technology.

ITSS has consulting, application development, package implementation and application support capabilities in various functional areas such as MRO, Telemetric, SCM, CRM, PLM, e-Business, HR, Finance, Engineering Analysis (Mechanical), etc. ITSS teams work on a wide range of technologies and have demonstrated capabilities to deliver total solutions, right from concept to deployment phases, and to provide application support services to Honeywell businesses. ITSS vision is to enable world

class business performance through delivering information and information services that are customer driven, accurate, timely, secure and cost effective. It servers SBGs like Aerospace, Automation and Control Solutions, Specialty Materials, Transport Systems, Corporate IT and Global Technology Services.

1.3 PROBLEM STATEMENT:

The main objective is to support plant-to-business integration (using Microsoft technologies) without reliance upon a specific technology, such as Oracle Interconnect or web Methods.

Instead, an XML-based approach is favored, using HTTP as the communications protocol for information delivery. This complements the company's internal development technologies (essentially an extension of Microsoft's .NET platform); ensures independence from a specific middleware vendor; and exploits the de-facto standard for web-based document exchange: XML.

The need for ERP Integration is as strong as ever in the market. Other integration issues are also becoming critical to customers such as E-Business, Supply Chain Optimization. More sophisticated customers are looking for a common solution for all major integration problems. Organization's role must be to continue to offer Uniformance® to ERP integration (full solution) or Support the customer's choice of corporate middleware.

XML is the de-facto standard for data integration both between applications and within applications but XML by itself is not integration .The Experion Application Framework (infrastructure for web-application development) is used to develop Business.FLEX web applications and hosts Workcenter HMI Web displays which includes Server/Station and makes extensive use of XML including "data access services" that expose data as XML streams which utilizes Microsoft's .NET platform.

Typical Integration Objectives:

- Send production targets from ERP; monitor progress during production; collect actual quantities of materials consumed and produced.
- Inventory Synchronization such as transfer official quantities for shipments and receipts from ERP system to plant yield accounting system. Calculate and validate plant material balance and report quantities of materials produced and consumption back to ERP.
- Improved Maintenance.

HPS Application Needs from an EAI application

- Need for Business Solutions example Business. FLEX to integrate with ERP systems and other 3rd party systems
- Need for Organizational solutions to integrate among themselves
- Need for such integration to be done by project engineering
- Industry standards to be used to integrate with ERP systems

1.4 WORK ENVIRONMENT

The various hardware and software description about the system used for the completion of the project is specified below.

1.4.1 HARDWARE SPECIFICATION

Processor	: Intel Pentium Dual Core (2.8GHz)
RAM	: 1 GB
HDD	: 80 GB

1.4.2 SOFTWARE SPECIFICATION

Operating system	: Windows XP Professional
Development Languages	: C#, XML, CSS
Database	: SQL Server 2005

IDE	: Microsoft Visual Studio 2005
Platform	: Experion Application Framework

(Since XML Services runs on the Honeywell Experion Application Framework, this EAF platform is very essential).

1.5 ROLE OF LANGUAGES AND TECHNOLOGIES USED

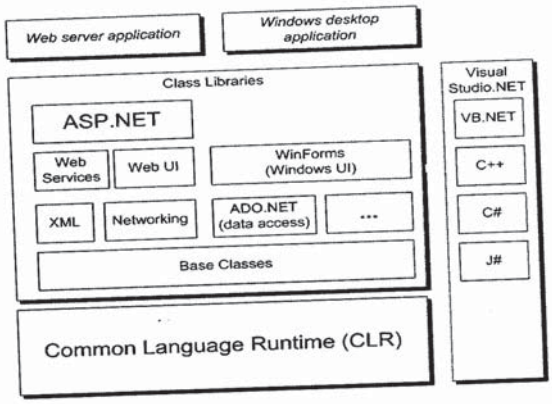
This section describes the various languages, tools and technologies used to develop this application.

NET FRAMEWORK

The .Net framework is the infrastructure for the new Microsoft .Net platform. This is a common environment for building, deploying and running web services and web applications. This framework is an integral windows component that supports building and running the next generation of applications and XML web services. It contains common class libraries and supports C++, C#, Visual Basic, script and COBOL. The .Net framework is designed to fulfill the following objectives.

- To provide a consistent object oriented programming environment whether object code is stored and executed locally, locally but internet-distributed or executed remotely.
- To provide a code execution environment that minimizes software deployment and versioning conflicts and that promotes safe execution of code, including code created by an unknown or semi trusted third party.
- To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
- To make the developer experience consistent across widely varying types of applications such as windows based applications and web based applications.

The elements of the .Net framework can be given in the following figure



The class library, the other main component of the .Net framework is a comprehensive object oriented collection of reusable types that you can use to develop applications ranging from traditional command-line or graphical user interface applications to applications based on the latest innovations provided by ASP.Net such as Web forms and XML Web services.

The .Net framework offers a number of benefits to developers.

- A consistent programming model
- Direct support for security
- Simplified development efforts
- Easy deployment and maintenance

ADVANTAGE OF .NET:

- simple and faster system development
- rich object model
- enhanced built in functionality

- Many different ways to communicate with the outside world
- Integration of different languages into one platform
- Easy deployment and execution
- Wide range of scalability
- Interoperability with existing applications
- Simple and easy to build sophisticated development tools
- Fewer bugs

C# .NET

C# is an object-oriented programming language developed by Microsoft as part of the .NET initiative and later approved as a standard by ECMA and ISO. C# combines the high productivity of Rapid Application Development (RAD) languages and the raw power of C++.

Visual C# .NET is part of a suite of products, called Visual Studio .NET, that also includes Visual Basic .NET, Visual C++ .NET, and the JScript scripting language. All of these languages provide access to the Microsoft .NET Framework, which includes a common execution engine and a rich class library. It combines the power and efficiency of C++, the simple and clean design of Java and the language simplification of Visual Basic.

It includes strong type checking, array bounds checking, and detection of attempts to use uninitialized variables, source code portability, and automatic garbage collection. It is intended for use in developing software components that can take advantage of distributed environments. It is suitable for writing applications for both hosted and embedded systems, ranging from the very large that use sophisticated operating systems, down to the very small having dedicated functions.

XML(Extensible Markup Language)

Extensible Markup Language (XML) is a cross-platform, extensible, and text-based standard for representing data. It is also a key technology in the development of Web services. It is a data wrapper allowing both dynamic and static data, structured and semi-structured data to cross system borders unlike relational databases that deal with static and structured data. XML has received widespread support and adoption in the computer industry because of its simplicity, extensibility, interoperability, and flexibility, all of which stem from its power to represent data independent of programming language, platform, or operating system.

XML Parsers

An XML parser takes as input a raw serialized string and performs certain operations on it. First it checks the syntactic well-formed ness of the XML data, making sure that the start tags have matching end tags and that there are no overlapping elements. Most parsers also implement validation against the Document Type Definition (DTD) or the XML Schema to verify that the structure and content are as you specified. Finally, the parsing output provides access to the content of the XML document via programmatic APIs. There are two popular XML parsing techniques for Java:

1. Document Object Model (DOM)
2. Simple API for XML (SAX)

The Document Object Model is a platform- and language-neutral interface that will allow programs and scripts to dynamically access and update the content, structure and style of documents. The document can be further processed and the results of that processing can be incorporated back into the presented page. Xerxes has full support for DOM and Simple API for XML (SAX). DOM Parsing is a tree-based parsing technique that builds up an entire parse tree in memory. Information from XML documents is read by invoking methods on various objects it contains.

The advantages of DOM Parser are:

- Random Access to widely separated parts of the document
- Read-Write API

SQL Server 2005 Overview

Microsoft SQL Server is a relational database management system (RDBMS) produced by Microsoft. Its primary query languages are ANSI SQL and Transact-SQL. It includes native support for managing XML data, in addition to relational data. For this purpose, it defines an `XML` data type that could be used either as a data type in database columns or as literals in queries. XML columns can be associated with XSD schemas; XML data being stored is verified against the schema. XML is converted to an internal binary data type before being stored in the database. Specialized indexing methods were made available for XML data. XML data is queried using XQuery; SQL Server 2005 added some extensions to the T-SQL language to allow embedding XQuery queries in T-SQL. In addition, it also defines a new extension to XQuery, called XML DML that allows query-based modifications to XML data.

1.6 ABOUT THE PRODUCT

Xml Services product is the product that integrates Organization's applications and ERP applications. The main benefits for applications using XML Services are integration of Organization's applications with third party systems and application integration productivity gains from utilizing common services. Xml Services is developed for industries including refining and petrochemicals, oil and gas, pharmaceuticals, chemicals, consumer goods, life sciences industries, power, mining, metals and minerals, and pulp, paper and printing.

The system focuses on the development of a common and generic application framework to build various applications of business groups in Organization. The generic components involve Modeler, Messaging Configuration and Tools.

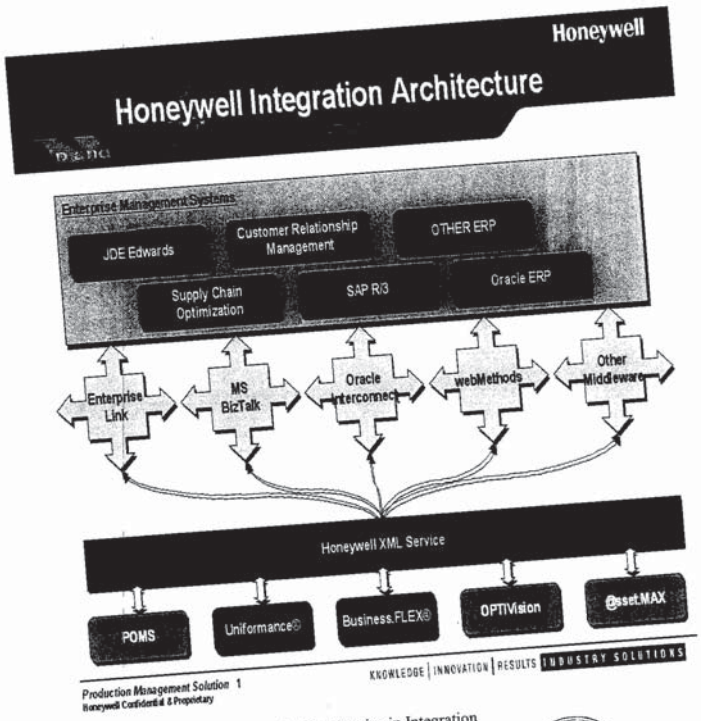


Fig. 1.1 Xml Service in Integration

1.6.1 MODELER

Modeler provides a single logical view for data located in diverse data sources. It allows user to read data from multiple different data sources and store data in different

data sources. Modeler is the graphical user interface to the functionality provided by WPKS Storage Service. It allows users to map either UML models or W3C XSD schemas or SQL Annotated schemas to data source elements.

The primary goal of the XML Service Modeler is to provide a user interface to map database fields to elements within an XML Schema. Working with EAF Data Access Services, it allows users to read from multiple databases and structure the output in the XML schema format specified by the user. Working with EAF Data Access Services, it allows users to write data to multiple databases based on the XML input provided. In combination with other EAF components, XML Service Modeler and EAF Data Access Services provide a secure, robust and efficient data access mechanism.

1.6.2 MESSAGING CONFIGURATION

Messaging Configuration configures a message to be an input to the BizTalk Server which will give the map results in xml file.

1.6.3 TOOLS

This module gives us tools to test the modeler map got as an output of the modeler.

CHAPTER 2

XML SERVICES MODELER INTRODUCTION

In this chapter an introduction about Xml service modeler is described. This deal with the introduction about what this tool is designed to and what all features it has.

2.1 MODELER INTRODUCTION

This project aims at the development of a XML Service component called Modeler. Modeler enables accessing different databases like SQL database, Oracle database, OLE DB, etc., under one platform at the same time. It supports using different DB functionalities like functions, stored procedures, views, etc.

Modeler maps different elements of the databases used, to certain elements/attributes. We can query, modify and update the databases with one common approach. The main advantage of Modeler is its simplicity in accessing all the databases at the same time. Honeywell has patented this product.

XML Modeler has the following features:

1. The primary goal of the XML Service Modeler is to provide a user interface to map database fields to elements within an XML Schema.
2. Working with XML Services, it allows users to read from multiple databases and structure the output in the XML schema format specified by the user
3. Working with XML Services, it allows users to write data to multiple databases based on the XML input given to Storage.
4. In combination with other EAF components, XML Service Modeler and EAF Storage provide a secure, robust and efficient data access mechanism.

2.2 MODELER BASICS

A number of terms used in the context of the Modeler are explained in this section. This section also contains details for configuring data source aliases using EAF Storage console.

The modeler is the heart of the product which controls various activities happening. So our current task is adding more features to the modeler to support even more database systems and other functionalities.

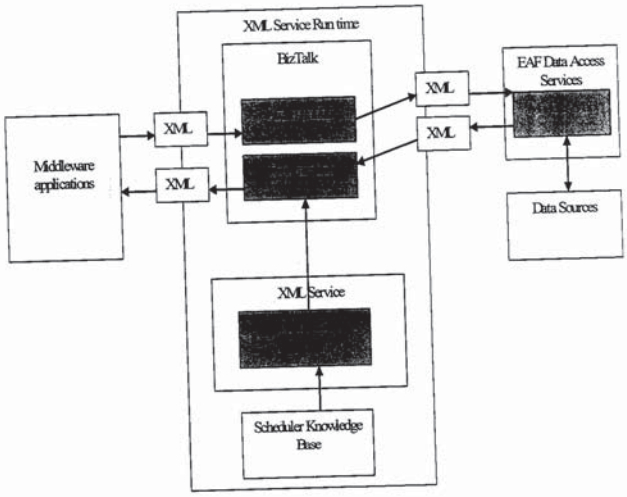


Fig. 2.1 Xml Service Overview

The following are Modeler specific terms used throughout this document. Each term is described in detail below

2.2.1 MODELER MAP

A Map is a term used to denote bindings between schema elements and database fields. A map is an entity that can only be created through the modeler. If the map reads data from underlying databases, it is called a Query Map. If it either modifies or deletes data from underlying databases, it is called an Update Map. A single schema can have both update as well as query maps. A modeler map is uniquely identified by its name and the namespace it belongs to. Namespaces are explained below.

2.2.2 NAMESPACE

A namespace in the modeler context refers to a set of logically related maps. An analogy in the Microsoft Windows environment is a directory that holds a set of files within it. A directory points to a set of files, similarly a set of maps can exist in the same namespace. Two different directories can have files with the same names; on the same lines two different namespaces in the modeler can have maps with the same name.

2.2.3 SCHEMA VIEW OR SCHEMA TREE

A modeler map has two distinct halves. One half always shows the xml tree hierarchy. This is referred to as either Schema Tree or Schema View in this document.

2.2.4 COMMAND VIEW OR COMMAND TREE

As stated above, a modeler map contains two distinct halves. The left half always displays the schema tree and the right half displays the commands tree. The commands tree shows the connections to data source aliases, creating SQL Statements etc

2.3 MODELER FEATURES

- Maps
  - Definition: definition about that particular DataSource
  - DST : DataSourceType mentions which kind of DataSource.
  - DSA: DataSourceAlias is physical Alias of that DST
  - Namespace: roof under which we keep all the Maps
- Data source
  - Query
    - IsPrimaryKey: let's you specify which is primary key
    - Static Value: which can be used throughout the Map
  - Modify: used to modify the database
  - Delete: used to delete from the Database
- Schema
  - Importing a schema: can import the schema which is created elsewhere
  - Creating a new schema: can create new schema by just right-click of mouse
  - data Type: can specify the data type of each element is schema
  - keyInParent: used to specify based on which element that child node loops
  - PostMethods: can specify to make a method execute right after that node
  - PreMethods: can specify to make a method execute right before that node
  - elementCondition: can set a method to check the element condition and then to execute
  - conditionContinue: this tells us whether to continue or not
  - forLoop: loops through the nodes till it satisfies the exit condition
- Local Parameters : can declare local parameters to use them in maps
- Global Parameters : can declare global parameters that can be used for dynamically passing output

- Direct Values: direct values are like static values that can be used inside the map scope
- Method (local, global, conditional method)
  - Local Method: can be used inside a particular map only.
  - Global Method: can be used in any map.
  - Conditional Method: conditional method checks for some condition and returns Boolean value
  - IsConstructor: tells us whether that particular method is constructor or not
  - Dependent Data Source: dependent data source tells, on which data source it is dependent
- Modeler Settings: can change the connection type
- Key: key which is used to sign the assemblies

2.4 EAF (EXPERION APPLICATION FRAMEWORK)

EAF is an application development and runtime environment that enables tighter integration between applications and Experion. The underlying platform for the XML Service(S) will be Honeywell's EAF environment, and will take advantage of many inherent EAF functions (such as storage services). The XML service will also require the accelerated development of some EAF functions that were originally planned for future EAF releases.

Using EAF for applications will have the following benefits:

1. Level of integration with the Experion™ DCS that can't be matched by 3rd parties.
2. A common look and feel for Honeywell applications.
3. Productivity gains by applications using common services provided by the framework e.g. Experion™'s extensive open integration infrastructure for operation over 3rd party systems.
4. Better integration between Honeywell Applications.
5. Many more like Security, Scalability and Availability etc.

2.5 EAI (ENTERPRISE APPLICATION INTEGRATION)

EAI can be used for different purposes:

1. Data (information) Integration: ensuring that information in multiple systems is kept consistent.
2. Process Integration: linking business processes across applications.
3. Vendor Independence: extracting business policies or rules from applications and implementing them in the EAI system, so that even if one of the business applications is replaced with a different vendor's application, the business rules do not have to be re-implemented.
4. Common facade: An EAI system could front-end a cluster of applications, providing a single consistent access interface to these applications and shielding users from having to learn to interact with different applications

2.6 INTERDEPENDENCIES OF THE APPLICATION

This product is dependable on the Experion™ Application Framework which is the platform over which XML Services works on. Hence this Framework acts as a prerequisite for this application.

CHAPTER 3

SYSTEM REQUIREMENTS

This chapter deals with the detailed specification of the requirements for this project and the main features that the product must be having. These requirements are well validated and using them test cases are also designed to use in the testing phase.

3.1 PRODUCT PERSPECTIVE

Xml Service product is the product that integrates Organization's applications and ERP applications. The main benefits for applications using XML Service are integration of Organization's applications with third party systems and application integration productivity gains from utilizing common services. The system focuses on the development of a common and generic application framework to build various applications of business groups in Organization.

3.2 USER INTERFACE

1. The need for ERP Integration is as strong as ever in the market. Send production targets from ERP; monitor progress during production; collect actual quantities of materials consumed and produced.
2. Inventory Synchronization such as transfer official quantities for shipments and receipts from ERP system to plant yield accounting system. Calculate and validate plant material balance and report quantities of materials produced and consumption back to ERP.
3. Improved Maintenance

3.3 MEMORY CONSTRAINTS

XML-SERVICES make use of various BizTalk server orchestrations. So high speed RAM is required. And to store the Maps in ODL directory it is simple to have 40GB Hard disk.

3.4 PRODUCT FUNCTIONS OVERVIEW

Send production targets from ERP.

Monitor progress during production.

Inventory synchronization.

Collect actual quantities of materials consumed and produced.

Improved maintenance.

3.5 CONSTRAINTS

The limitations on XML-SERVICES option

- Only XML can be used as data across all applications.
- XML-SERVICES needs EAF to run
- XML-SERVICES needs Data Access Service of EAF.

3.6 SPECIFIC REQUIREMENTS

3.6.1 FUNCTIONAL REQUIREMENTS

Functional requirements define the fundamental actions that take place in the software in accepting and processing the inputs and in processing and generating the outputs.

These include:

- *Data source Type Creation*
- *Data source Access Creation*
- *Namespace Creation*
- *Map creation*
- *Configuration with BizTalk server*

3.6.1.1 Functional Requirement 1:

The user must create Data source type in order to establish a typical connection with any of the databases.

3.6.1.2 Functional Requirement 2:

In order to access the data access service of EAF user must create the data source access connection.

3.6.1.3 Functional Requirement 3:

Namespace is needed for making a logical container for all Maps.

3.6.1.4 Functional Requirement 4:

Map creation is the last phase where each map represents the schema view of all databases.

3.6.1.5 Functional Requirement 5:

If we want to give the Maps to BizTalk configuration we have to configure Modeler Map created.

3.7 PERFORMANCE REQUIREMENTS

- It supports multiple users simultaneously.
- Effective integration with all ERP components.
- The tool takes XML as input format which is widely accepted.
- It takes nearly 8 megabytes of data and writes into the database in single query.
- Provides a central control for entire database.
- The benefit of this tool is this provides the great user interface to fetch or write the data into database and also it is loosely coupled to the other enterprise software.

3.8 SOFTWARE SYSTEM ATTRIBUTES

3.8.1 RELIABILITY

This system provides a reliable and secured way of data flow.

3.8.2 AVAILABILITY

The system is available 24/7. The server will be always on for receiving the messages. Proper assistance is provided if any system failure occurs. The high back up mechanism provides redundancy but ensures availability of data. The system shall allow users to restart the application after failure with the a little loss of data.

3.8.3 SECURITY

1. Keep specific log or history.
2. Assigning certain functions to different modules.
3. Checking the data integrity.

CHAPTER 4

SYSTEM LEVEL DESIGN

The basic design of the modeler is described in this chapter using context diagrams and use case diagrams and dataflow diagrams. Input and the expected outputs are being shown in this chapter.

4.1 CONTEXT DIAGRAM

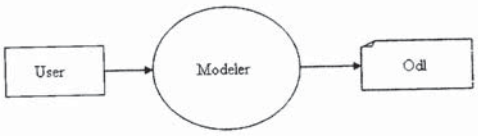


Fig. 4.2 Context Diagram

This context diagram talks about the user interaction with the modeler and the output that the modeler is going to give. Odl is a folder that will be containing the dll that is going to be generated when we click the configure button and the xml file contains the mapping information of the schema.

4.2 DATAFLOW DIAGRAM (LEVEL1)

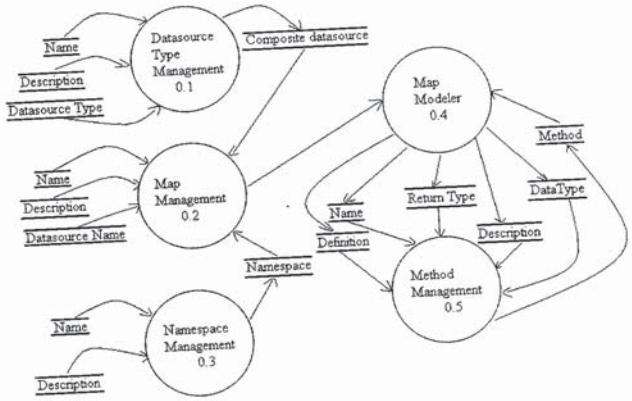


Fig 4.3 Modeler Design

This diagram talks about the flow of the data between each module of the tool. First we pass the name, description and the data source type for creating the composite data source. And then for creating the map we need to pass the name of the map and description and data source name and which opens the map window, where we can create the maps of our own, which is map modeler. And in Namespace management the name and the description of the namespace are being created. In modeler map every input from the previous ones. In this we create the maps which maps the schema tree to the database schema.

4.3 DATAFLOW DIAGRAM (LEVEL 2)

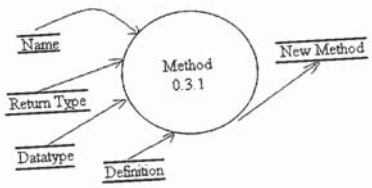


Fig. 4.4 Method Design

Method is to create the method that can be used inside the maps, to which we pass the name and the return type and the data type and definition.

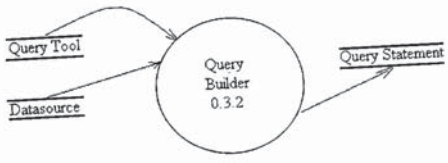


Fig 4.5 Query Builder Design

Query builder is a tool which can be used to create the queries that fetches the data from the database

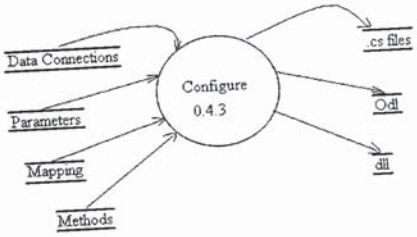


Fig. 4.6 Configure

Configure is used to compile the maps which generates the .cs files and the odl and the dll.

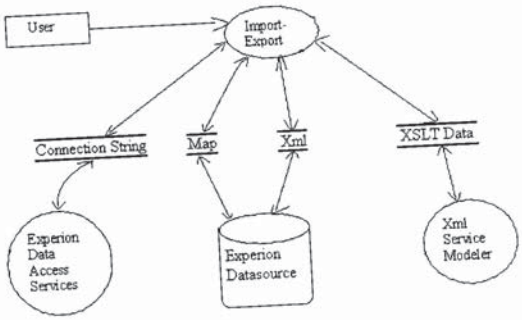


Fig. 4.7 Import Export

Import and export is to import and export the maps that are being created from one machine to another without creating them again.

4.4 USECASE DIAGRAMS

4.4.1 Database Type Management

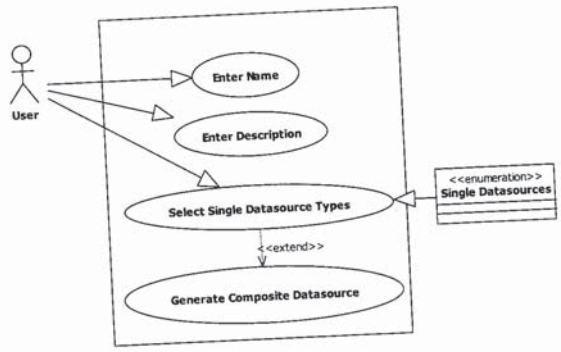


Fig. 4.8 Database Type Management

In this use case user enters the name of the data source and the description and the number of data source types

4.4.2 Namespace Management

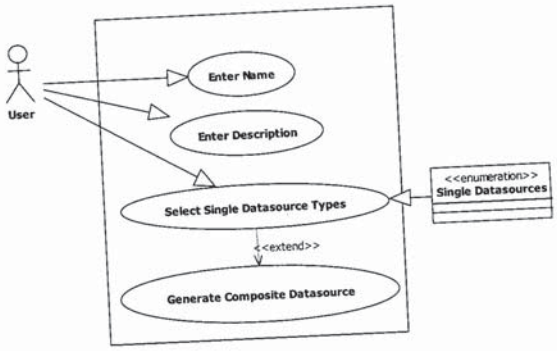


Fig. 4.9 Namespace Management

4.4.3 Import Export Tool

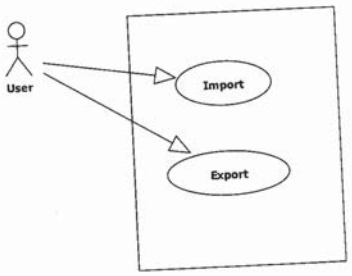


Fig. 4.10 Import Export Tool

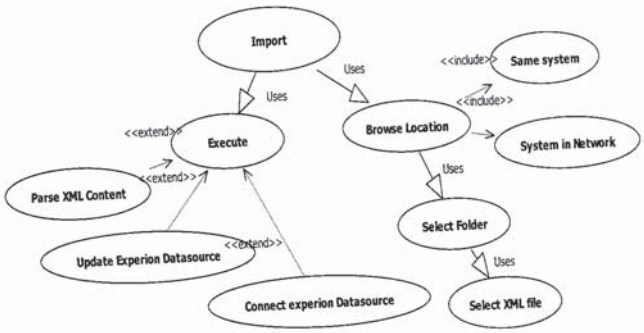


Fig. 4.11 Import Usecase

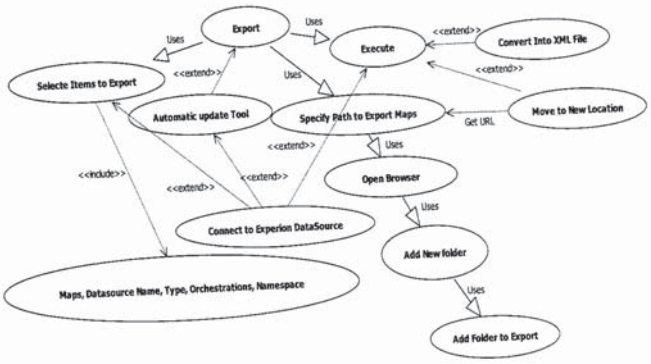


Fig. 4.12 Export Usecase

4.5 TABLE STRUCTURES

1. DATASOURCE TYPE:

Column Name	Data Type	Length
DST_Type	Nvarchar	50
Description	varchar	100
DST_Name	Nvarchar	50

2. DATASOURCE NAME:

Column Name	Data Type	Length
DSN_Name	Nvarchar	50
DSA_Name	Nvarchar	50
User_DSN_Name	Nvarchar	100
DST_Name	Nvarchar	50
Description	varchar	100
DST_Type	Nvarchar	50
Location	Nvarchar	50

3. DATA SOURCE ACCESS:

Column Name	Data Type	Length
DSA_Name	Nvarchar	50
Namespace_Name	Nvarchar	50
DSN_Name	Nvarchar	50
Description	varchar	100

4. MAP DETAILS:

Column Name	Data Type	Length
Name	varchar	100
Namespace_Name	Nvarchar	50
Description	Nvarchar	50
DSA_Name	Nvarchar	50
DST_Name	Nvarchar	50
Date_Created	Nvarchar	50
Time_Created	Nvarchar	50
Configured	Boolean	

5. USER CREDENTIALS:

Column Name	Data Type	Length
DB_UserName	varchar	10
DB_Password	Nvarchar	10
DSN_Name	Nvarchar	50
DML_Execute	Boolean	
DDL_Execute	Boolean	
View	Boolean	

4.6 DATA DICTIONARY

1. Composite Datasource Type: Datasource containing different types of databases

2. Namespace: Container

3. Name: String

4. Datasource Name: String

5. Description: String

6. Datasource Type: Standard Datasources

7. Datatype: C# datatype

8. Return Type: Datatype

9. Odl: Map file

4.7 STRUCTURED CHART

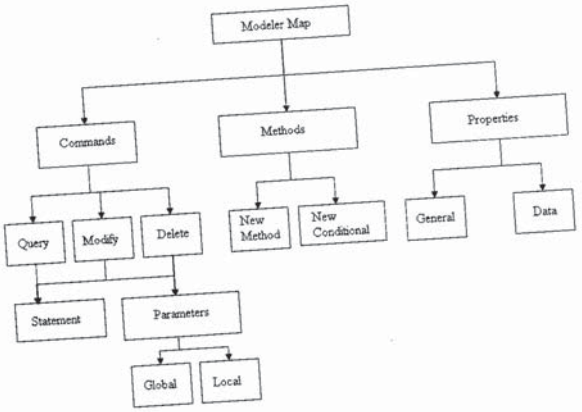


Fig. 4.13 Structured Chart -Modeler Map

In this structured diagram, the broad picture of the modeler is being explained. It has three main tabs commands and the methods and the properties. In commands we select the type of the map that we are going to create i.e…, query, modify, delete and in this we select the statement and then local and global parameters.

CHAPTER 5

XML SERVICES MODELER IMPLEMENTATION

These are the three main modules that are there in the XML Service

1. Modeler.
2. Messaging Configuration.
3. Tools.

Modules in Modeler are

1. Datasource Type Management
2. Namespace Management
3. Map Management
4. Method Management
5. Modeler Configuration

5.1 DATA SOURCE TYPE MANAGEMENT

This is used to create the data source type .That data sources from that data source type is used in the maps to retrieve the tables .And that tables data is accessed using the modeler. The modeler generates output in xml format. The screen shot for this is shown in below.

5.1.1 Create Data Source in EAF

Data sources are created in EAF. (Experion Application Frame work) .The screen shot for this is shown in the figure. EAF is the heart of the XML Service, it is has all the information about the data sources that are being used in XML Service. It has the User credentials and Role Credentials that are used to connect the Database. It has an advantage of loose coupling and when ever we need to change the connections to

atabase we can just change the connections in EAF rather than changing in XML ervice and re doing everything.

5.2 NAMESPACE MANAGEMENT

5.2.1 Create Name Space In XmlService Modeler

This is used to logically organize maps in particular name space. It requires Datasource access to be created first

5.3 MAP MANAGEMENT

5.3.1 Create Map Using Map Management

After clicking on Map Management in configuration explorer, management window opens. We can create map by specifying the Datasource type, Datasource access, and Namespace and map description.

5.4 METHOD MANAGEMENT

In Method Management the, we can create the maps that are exactly looks like the methods in c#, this form validates the syntax as it is in c#. we can create the parameters and also we can make it as either a global parameter or a constructor.

When you right click on a statement .select a new statement it will open a window shown below. Here it opens a window. Using this you can automatically generate a SQL statement by selecting the tables fields.

.4.1 Functionalities of Map modeler

This is another important module where the products entire functionality resides, this is where we can create the maps and then configure them according to the requirement. First we need to select the data source and then the table from the selected connection and then we need to select the required columns and we can generate the query using query builder wizard.

Create some elements on the left side of the modeler map. This is called schema tree. Map the elements from command tree to schema tree. Next click on configure map .If there are no errors it displays map configured successfully.

ODL file contain xml schema file, that contain entire information about the data source Connection and all the details.

Schema file will contain description of the ODL file.

When you click on configure button this ODL file will be converted to .cs file using XSLT transformations. When you compile the .CS file the DLL is generated.

This DLL is used in Test Client to see the results.

5.5 MESSAGING CONFIGURATION

Messaging configuration is the module which makes this entire xmlservice a better product than any other product. This product is on top of the BizTalk Server 2006. It uses the BizTalk Server features and there is a scheduler in this module, which schedules the messages to a particular file port that lets the orchestration to be kicked off. The

rchestration executes the map and fetches the results from the database and then it puts ı another file port.

In this we select the Templates that are there in this module. There are three templates Get, Set, GetSet. Get is to get the results, Set is to inserts into the database, GetSet is to get the results and to insert the results.

In the Get Wizard we specify the Input message schema and also output schema. And data source alias and the question name and in the project details form we specify the key name and press finish.

Then it opens the orchestration in visual studio and then we need to build the solution then deploy it. Then open the BizTalk administrator and the start the orchestration and then keep the xml file in the receive port, which kicks off the orchestration. And the orchestration executes the dll and fetches the results and puts the xml file in the send port.

5.6 IMPORT EXPORT WIZARD

This is a tool that imports and exports the maps from one machine to another, this tool is developed by myself. The main functionality of this tool is to read the .cs files and the .dll files from one machine to and also the database entries of that particular maps and then has to import back into the another machine.

CHAPTER 6

TESTING

Testing is an important part of the software development life cycle. The amount of testing required is related to the size and complexity of the application. Testing mainly focuses on the identifying the unseen defects in the developed software. There are different types of testing which are discussed in this section.

TYPES OF TESTING:

6.1 UNIT TESTING

Unit testing concentrates on each unit of the software implemented in the source code. The developer at the module level carries this out and this testing is white-box oriented. The module interface is tested to ensure that information properly flows into and out of the program unit under test. All independent paths through the control structure are exercised to ensure that all statements in a module have been executed at least once.

6.2 VALIDATION TESTING

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software tests called validation testing begins. Software validation is achieved through a series of black box tests that demonstrate conformity with requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with requirements. Both the plan and procedure are designed to ensure that all functional requirements are satisfied, all performance requirements are achieved, documentation is correct and other requirements

re met. After each validation test case has been conducted, one of the two possible onditions exists. The function or performance characteristics conform to specification nd are accepted (or) A deviation from the specification is undiscovered and a deficiency ist is created.

6.3 INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure, while at the same time conducting tests to uncover errors associated with interfacing. That is, the program is constructed and tested in small segments, which makes it easier to isolate and correct. This testing focuses on the design and construction of the software architecture. The testing team carries this out and this testing is black box oriented.

6.4 SAMPLE TEST CASES

TESTCASES FOR VALIDATION TESTING:

S No	Test Cases	Valid User	Cycle1
1	Registered Users try to enter into Xml Services Configuration Studio with their Role Credentials.	Yes	Pass
2	When Xml Admin tries to change the schema used in the Map Modeler.	Yes	Pass
3	Storage Admin tries to create map in Map Management of XML Modeler without DST and DSA	Yes	Pass

Table 6.1 Sample Validation test cases and its results

TEST CASES FOR UNIT TESTING:

S No	Test Cases	Expected result	Cycle1
1	Will it able to configure several maps under single namespace?	Yes	Pass
2	Will it able to modify a common data in all the datasources listed in tree view?	Yes	Pass
3	Querying all the database tables with a common query builder tool	Yes	Pass
4	Will the user be able to modify the connection string once DST is created in Xml Services tool?	Yes	Pass
5	Is there any option for the admin to delete the map after it is configured to datasources?	Yes	Pass
6	Is the application providing option to create composite data source for multiple databases?	Yes	Pass
7	Will the tool able to upload already existing schema files as input to database tables?	Yes	Pass

S No	Test Cases	Valid User	Cycle1
8	Will the tool able to create any common methods for Xml schemas?	Yes	Pass
9	Can we pass the maps created in Xml Services as input to BizTalk orchestration?	No	Pass
10	Will it able to modify a common data in all the datasources listed in tree view?	Yes	Pass

Table 6.1 Sample Validation test cases and its results

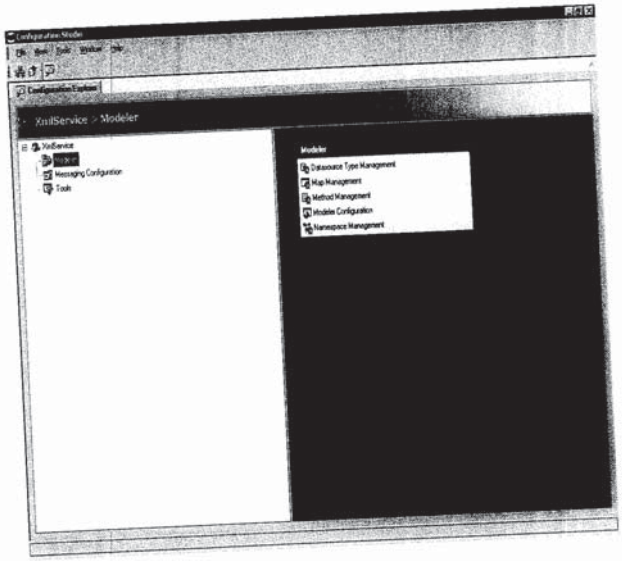
CONCLUSION

Xml Service is a tool that can fetch and write data into different databases at a single point of time. And this tool can take input as an xml format, which is accepted as a global format of data exchange. And it takes around 8Megabytes of data and writes it into the database in a single query. As this can read and write data into different database it can used in the integration of the organization. The benefit of this tool is this provides the great user interface to fetch or write the data into database and also it is loosely coupled to the other enterprise software, so it can provide the ease of changing some part of the code without changing the entire code.

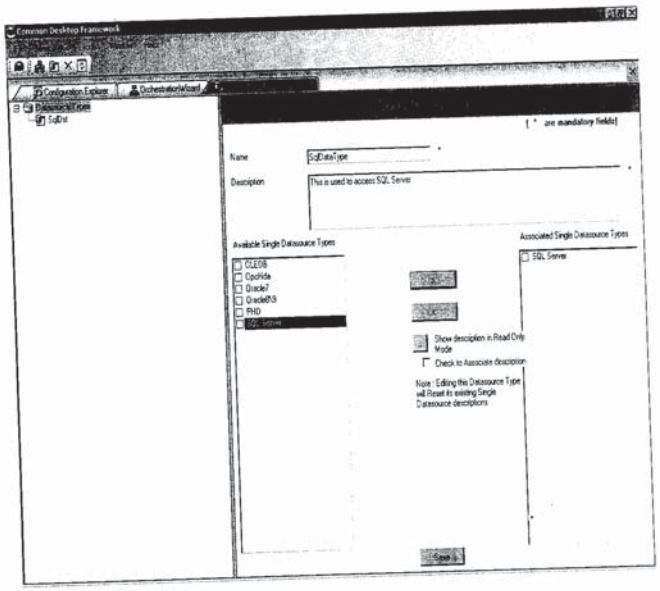
APPENDICES

SCREEN SHOTS:

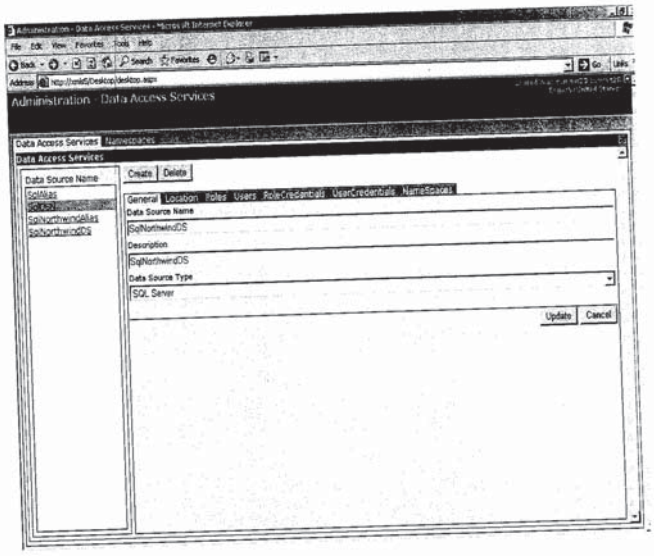
1. XMLServices Configuration Explorer showing Modeler:



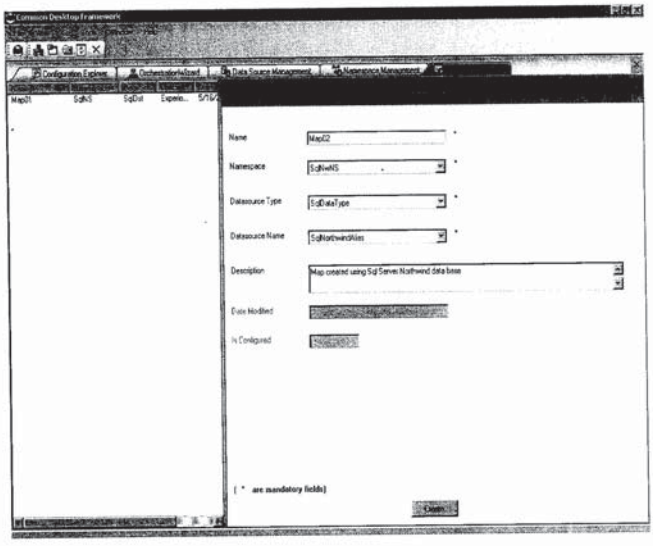
2. Data Source Type Management:



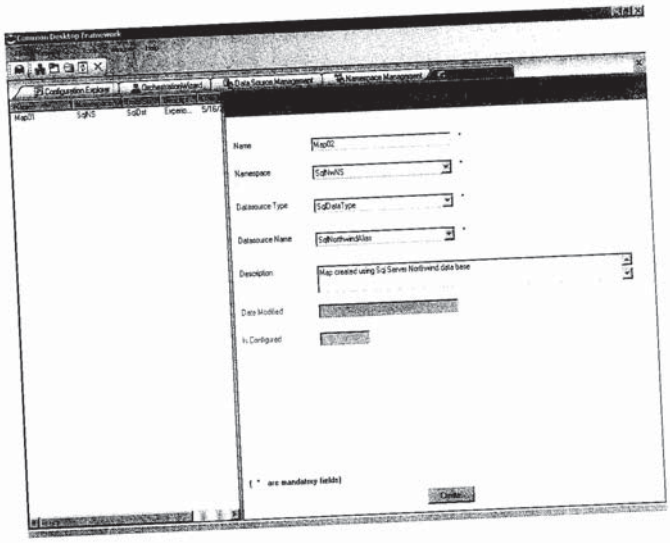
3. Experion Desktop:



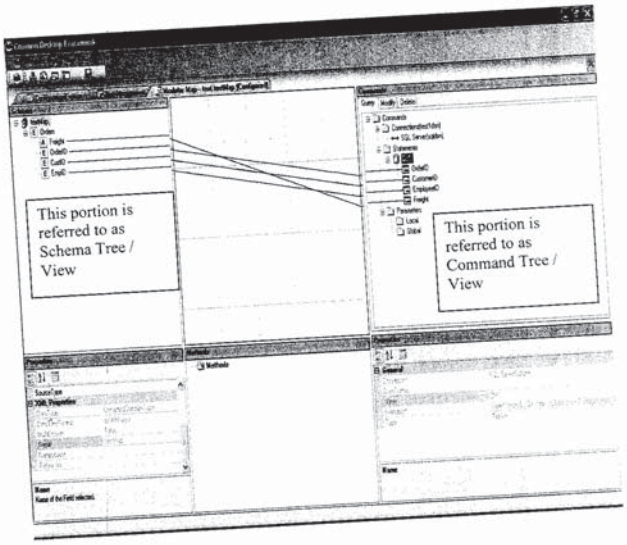
4. Map Management:



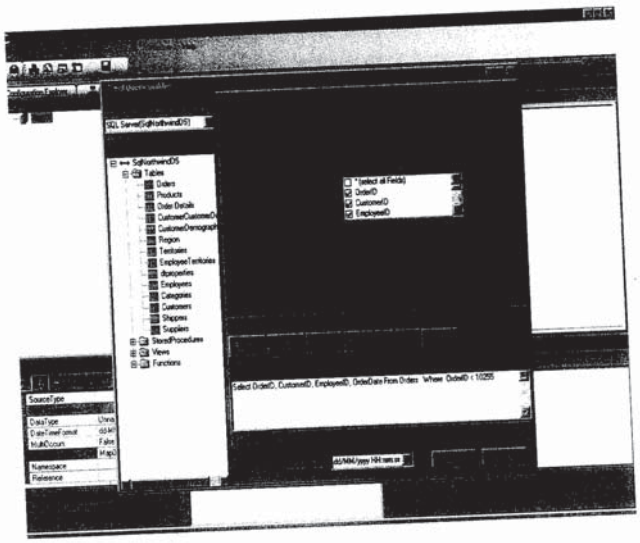
5. Creating a Map Using Map Management:



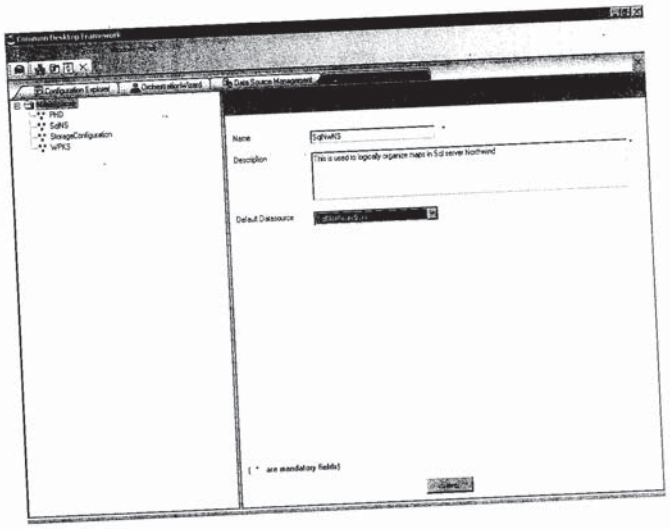
6. Modeler Showing Schema and Command Views:



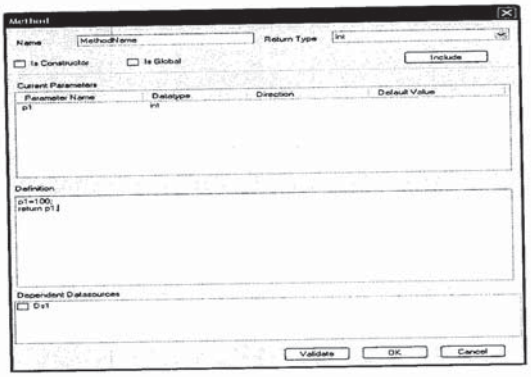
7. SQL Query Builder:



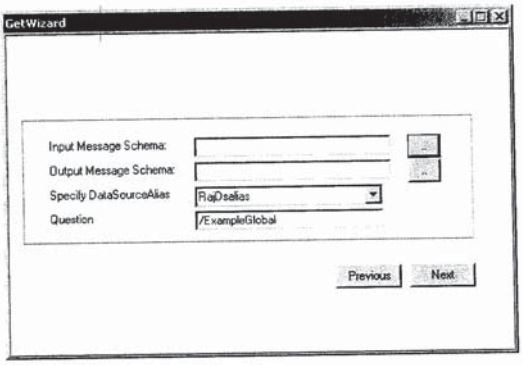
8. Namespace Management:



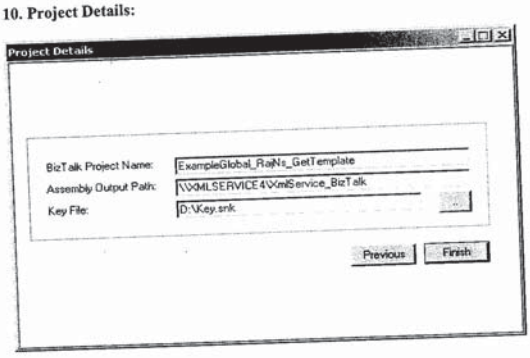
8. Method Management:



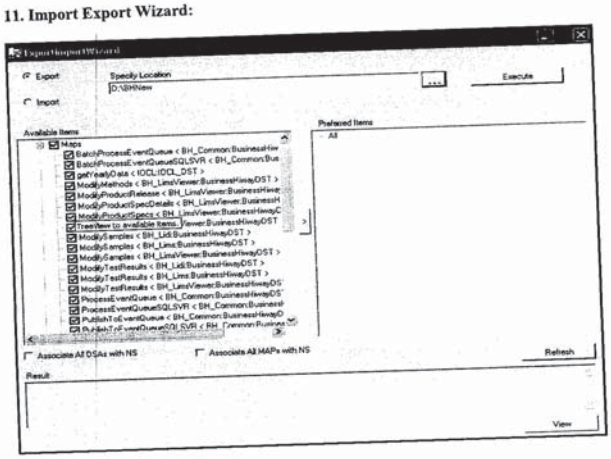
9. Get Wizard:



10. Project Details:



11. Import Export Wizard:



REFERENCES

Book Reference:

1. Heather Williamson., "XML: The Complete Reference", Tata McGraw-Hill Publishing Company Limited, 2001
2. Jain, V.K., "The Complete Guide to C# Programming", 2003
3. Simon Robinson., "Professional C#", Wiley Publishing, Indianapolis, 2004
4. Steven Holzner., "Teach Yourself XML in 21 days", Third Edition, Pearson Education. , 2002

Online Reference:

5. www.codeproject.com
6. www.csharpconer.com
7. www.microsoft.com/.net/basics
8. www.w3schools.com