ENTERPRISE INFORMATION INTEGRATION USING SERVICE ORIENTED ARCHITECTURE

PROJECT REPORT

Submitted by

M.PRABHAVATHY

Register No: 0920108016

in partial fulfillment for the award of the degree of

of

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

APRIL 2011

KUMARAGURU COLLEGE OF TECHNOLOGY

(An Autonomous Institution Affiliated to Anna University of Technology, Coimbatore)

COIMBATORE – 641 049

Department of Computer Science and Engineering

PROJECT WORK

PHASE II

APRIL 2011

This is to certify that the project entitled

ENTERPRISE INFORMATION INTEGRATION USING SERVICE-ORIENTED ARCHITECTURE

is the bonafide record of the project work done by

M.PRABHAVATHY

Register No: 0920108016

of M.E. (Computer Science and Engineering) during the year 2010-2011.

V. P. Sumathi

Project Guide

Head of the Department

Submitted for the Project Viva-Voce examination held on 25/4/11

25/4/11

Internal Examiner

25/4/2011

External Examiner

DECLARATION

I affirm that the project work titled "**ENTERPRISE INFORMATION INTEGRATION USING SERVICE ORIENTED ARCHITECTURE**" being submitted in partial fulfillment for the award of M.E degree is the original work carried out by me. It has not formed the part of any other project work submitted for award of any degree or diploma, either in this or any other University.

M.

M.Prabhavathy
0920108016

I certify that the declaration made above by the candidate is true

Signature of the Guide,

V.P. Sumathi

Ms. V.P.Sumathi M.E.,
Assistant Professor
Department of Computer Science & Engineering,
Kumaraguru College of Technology,
(An Autonomous Institution)
Coimbatore.

ACKNOWLEDGEMENT

First and foremost, I would like to thank the almighty for enabling me to complete this project.

I whole-heartedly wish to express my special thanks to our Chairman, Management, and Director for giving this great opportunity to pursue this course. I thank **Dr. S. Ramachandran,** *Principal*, for providing me with the necessary facilities and Infrastructure to work on this project.

I would like to extend my thanks to **Mrs. P. Devaki, M.E., (Ph.D),** *HOD*, Department of Computer Science and Engineering for her valuable suggestions.

I express my heartiest thanks to my Project Guide **Mrs V.P.Sumathi,M.E.,** *Assistant Professor*, Department of Computer Science and Engineering for her valuable suggestions and support right from the beginning to perform my project work extremely well.

I thank our Project Coordinator **Ms. V.Subramani M.Tech** *Associate Professor*, Department of Computer Science and Engineering, for his valuable advice and guidance throughout this project.

I would like to convey my honest thanks to all **Teaching and Non-Teaching** staff members of the department for their support.

TABLE OF CONTENTS

Coimbatore - 641 049.

Department of Computer Science and Engineering

CSIR
Sponsored

National Conference
on
Service Oriented Computing

Certificate

certify that ~~Dr.~~/~~Prof.~~/~~Mr.~~/~~Mrs.~~/Ms........M. PRABHAVATHY..........

aguru College of Technology..............................has presented

ENTERPRISE INFORMATION INTEGRATION USING SERVICE ORIENTED ARCHITECTURE..........in the

on Service Oriented Computing(NCSOC 2011) held on 24th & 25th March

handrakala)
g Secretary

S Devaki
(Prof.P.Devaki)
HoD

(Dr.S. Rama
Princi

ABSTRACT

In today's fast running business world,its impossible to store all information in a single place ,single system as well as single platform.We need to store in a different system as well as different platform.Here interoperability is the main problem focusing here . So a Web Services based Service Oriented Architecture are the distributed technology that solve those interoperability problem. In order to integrate the data from heterogeneous data source,web service based service-oriented architecture is presented here. First,the characteristics of various data sources such as MySql,MS-Acess,Oracle,Spreadsheet should analyzed and their structure and semantics should be understood.Secondly we need to wrapped the data source into web service through muti-agent technology.Here we are using data federation as a technique and enterprise information integration(EII) as a technology to integrate data from various data source on on-demand basis.Using federated database splitting the user query into many sub-queries and given to various data source and combining the results and producing the single unified view of the data.Here the security and stability of source system are retained because of wrapping techniques.Web service based service-oriented approach is used here because of this characteristics such as interoperability,loosely coupling,reusability,composability etc.

ஆய்வுச்சுருக்கம்

மிக வேகமாக வளர்ந்துக் கொண்டிருக்கும் இவ்வுலகில், நம்மால் எல்லா தகவல்களையும் ஒரிடத்தில், ஒரே முறையில் சேகரித்து வைப்பது என்பது இயலாத காரியம். ஒவ்வொரு நிறுவனமும் பல்வேறு கிளைகளை வெவ்வேறான இடத்தில் அமைத்துள்ளது. ஒவ்வொரு கிளையும் தங்களுக்கு விருப்பமான முறையில் தகவல்களை சேமிக்கின்றன, பல்வேறான முறையில் அதாவது மைக்ரோசாஃப்ட் எக்செஃஸ், ஆரக்கிள், மை எஸ்கியூஎல், எக்ஸ்எம்எல் ஆகிய முறைகளில் தகவல்களை சேமிக்கின்றன. இந்த ஆய்வின் முலம்,நாம் ஒரிடத்தில் இருந்துக் கொண்டே பல்வேறு கிளைகளின் தகவல்களை அறியவேண்டும். இதற்கான தீர்வு சேவை அடிப்படையிலான கட்டமைப்பாகும். அதாவது சேவை அடிப்படையிலான கட்டமைப்பு என்பது இணையதளத்தின் மூலம் அறிவது என்பது கண்டறியப்படுகிறது.

மேலும் ஆய்வின் சிறப்பை அதிகப்படுத்துவதற்க்காக ஒரிடத்தில் இருக்கும் தகவல்களை பெருவதற்கான நேரமும், கிளைகளின் தகவல்களை பெருவதற்கான நேரமும் வரைபடத்தின் மூலம் ஒப்பிடபடுகிறது. இந்த ஆய்வின் முடிவில் கிடைக்கபடும் வரைபடத்திலிருந்து பழைய ஆய்வு முறைகளை விட இந்த ஆய்வு முறை சிறந்தது என கண்டறியப்படுகிறது.

LIST OF FIGURES

LIST OF ABBREVIATIONS

ETL	Extraction Transformation Loading
EAI	Enterprise Application Integration
EII	Enterprise Information Integration
SOA	Service Oriented Architecture
XML	Extensible Markeup Language
UDDI	Universal Discovery Discription Interface
WSDL	Web Service Description Language
SOAP	Simple Object Access Protocol

CHAPTER 1

INTRODUCTION

1.1 DATA INTEGRATION

Data integration is the process of combining data from a heterogeneous set of data stores to create one unified view of all that data. Data integration [1] involves combining data residing in different sources and providing users with a unified view of these data. Data integration joins two or more sets of data which actually reside at different sources for analysis and reporting.

1.1.1Data Integration Techniques

Basically three techniques are there:

a) Data Consolidation

b) Data Propogation

c) Data Federation

1.1.1.1 ETL

Data Consolidation [2] captures data from multiple source systems and integrates it into a single persistent data store. This data store may be used for reporting and analysis as in data warehousing, or it can act as a source of data for downstream applications as in an operational data store.

- ✓ With data consolidation, there is usually a delay, or *latency*, between the time updates occur in source systems and the time those updates appear in the target store. Depending on business needs, this latency may be a few seconds, several hours, or many days.
- ✓ The term *near real time* is often used to describe target data that has a low latency of a few seconds, minutes, or hours.
- ✓ Data with zero latency is known as *real-time data,* but this is difficult to achieve using data consolidation.

1.1.1.2 EAI

Data Propagation [3] applications copy data from one location to another. These applications usually operate online and *push* data to the target location; i.e., they are event-driven. Updates to a source system may be propagated asynchronously or synchronously to the target system. Synchronous propagation requires that updates to both source and target systems occur in the same physical transaction.

- ✓ Enterprise application integration (EAI) and enterprise data replication (EDR) are examples of technologies that support data propagation.

1.1.1.3 EII

Data Federation provides a single virtual view of one or more source data files. When a business application issues a query against this virtual view, a data federation engine retrieves data from the appropriate source data stores, integrates it to match the virtual view and query definition, and sends the results to the requesting business application. By definition, data federation always *pulls* data from source systems on an on-demand basis. Any required data transformation is done as the data is retrieved from the source data files. Enterprise information integration (EII) [4] is an example of a technology that supports a federated approach to data integration.

1.1.2 Data Integration Framework:

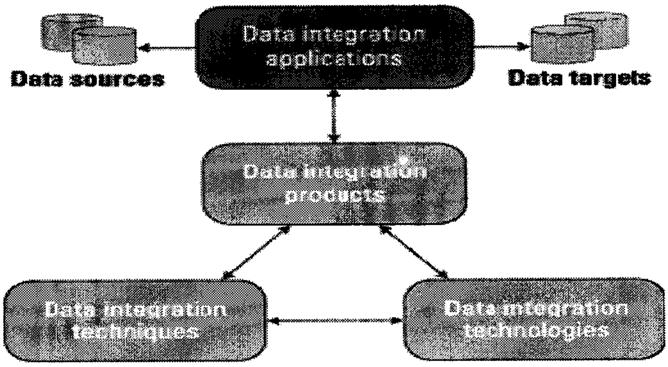

Data integration applications
Data sources
Data targets
Data integration products
Data integration techniques
Data integration technologies

Fig 1 Data Integration Architecture

1.1.3 Enterprise Information Integration

EII provides a virtual business view of dispersed data. This view can be used for demand-driven query access to operational business transaction data, a data warehouse, and/or unstructured information.

It is the process of integrating the information from heterogeneous data source.

- On-demand
- Single consolidate view
- The aim of effective integration, namely having the right information at the right place at the right time still remains largely unattained.
- Data within an enterprise can be stored in various formats: – relational databases (in a large varieties), text files, XML files, spreadsheets, each with their own indexing and data access methods
- Standardized APIs to retrieve and modify data from a generic data source:– ODBC, JDBC, OLE DB, and ADO.NET
- Standard formats for representing data within a file, the best-known is XML

1.1.4 How EII works ?

Applications address queries to the EII layer [5] . EII acts as a 'PULL' engine which waits for the requests, splits the query across multiple heterogeneous data source systems gathers transactional data sets, merges them together and then pushes them out to the requesting applications. These requesting applications can be a Web-Service, Excel or some other frontend.

1.1.5 Benefits of EII

Enterprise information integration's claim to fame is its ability to federate data. It provides a single point of access to disparate information sources. This reduces the complexity inherent in client applications attempting to join various data sources while offering another way to access that information. Some major benefits of using this technology are:-

- EII Shields applications from the details such as location and format of the information, protocols and query languages supported by the information sources, and the programming

- It also allows the applications to process data independent of changes to the underlying data management infrastructure.
- EII can act a virtual database that insulates the data warehouses from the impact of unmanaged raw queries
- Automate operations and data extraction from any type of data system: Web sources,relational databases, XML, Web Services, files (flat, PDF, Word, Excel, logs, etc.), data warehouses, pplications, etc.
- Support to SOA architectures, as data can be accessed through Web Services and predefined exchange formats can be generated (e.g. XML).

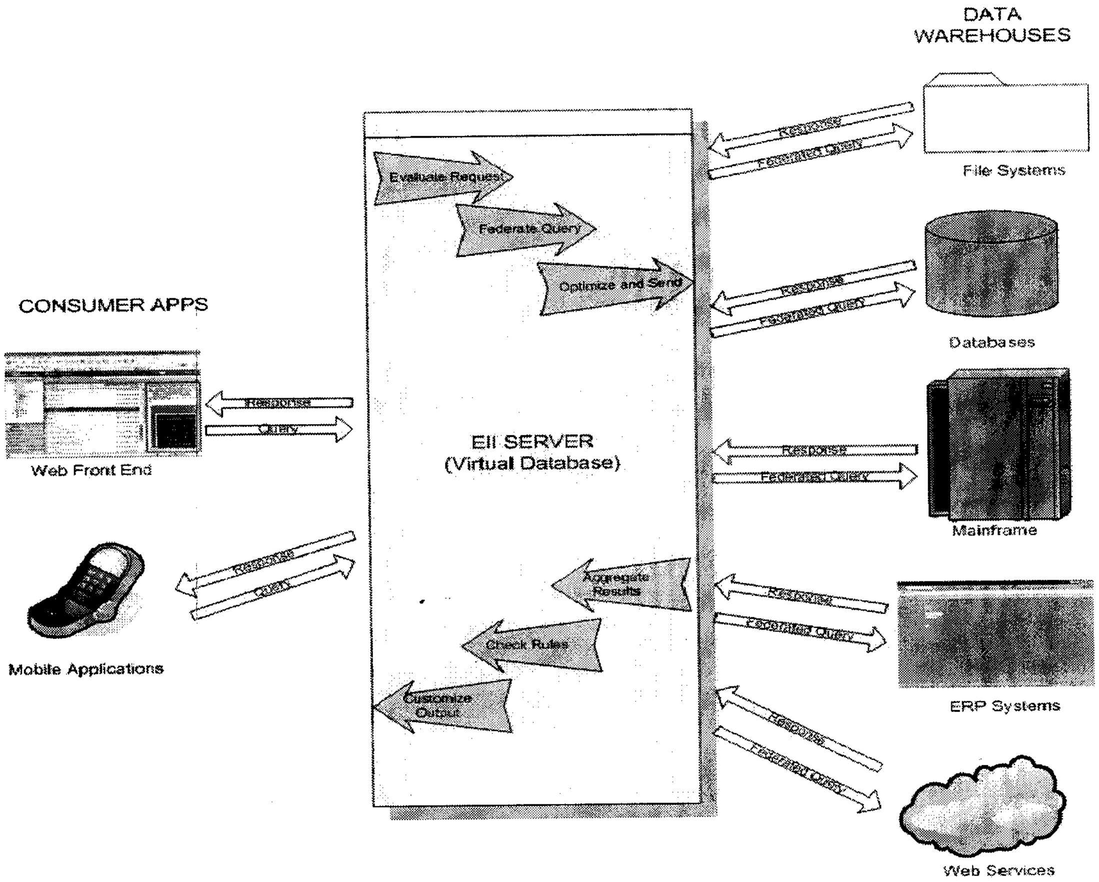


Fig 2.EII Architecture

.1.6 Limitation of EII

he EII model has a number of limitations. These are:-

- The fact that querying Data sources for reporting, without degrading their performance is still an issue. It is an especially important issue when the source systems are older legacy systems.
- EII also tends to suffer from data quality problems, to an even greater extent than the data warehouse model. Unlike data maintained in a data warehouse, which is extracted from operational sources and then standardized and cleansed, EII tools generate a virtual view of the data they assemble from various operational sources, which generally contains mismatched data.

1.2 SERVICE ORIENTED ARCHITECTURE:

Service-oriented architecture (SOA) [6] is a flexible set of design principles used during the phases of systems development and integration in computing. A system based on a SOA architecture will package functionality as a suite of interoperable *services* that can be used within multiple separate systems from several business domains.

1.2.1 What is SOA?

It's an architectural style of building software applications that promotes loose coupling between components so that you can reuse them. Thus, it's a new way of building applications with the following characteristics:

- Services are software components that have published contracts/interfaces; these contracts are platform-, language-, and operating-system-independent. XML and the Simple Object Access Protocol (SOAP) are the enabling technologies for SOA, since they're platform-independent standards.
- Consumers can dynamically discover services.
- Services are interoperable

1.2.2 Service oriented architecture:

- Service Provider
- Service Consumer

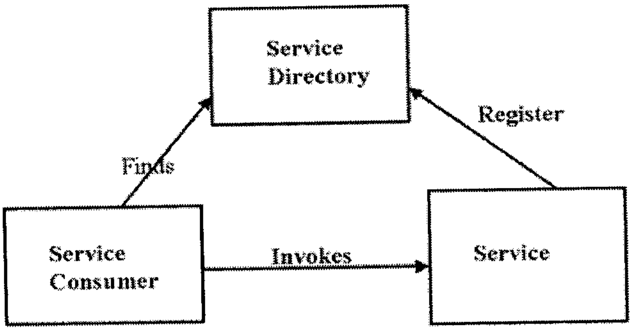


Fig 3 SOA Architecture

The basic building block of SOA is the *service*. A service is a self-contained software module that performs a predetermined task: "verify a customer's credit history," for example. Services are software components that don't require developers to use a specific underlying technology.

1.2.3 Common principles of service-orientation:

- services are loosely coupled
- services share a formal contract
- services abstract underlying logic
- services are composable
- services are reusable
- services are autonomous
- services are stateless
- services are discoverable

Services are Loosely Coupled

Coupling refers to a connection between two things..A fundamental principle of service-orientation is that units of logic that are classified as services retain a minimal level of coupling.

Services Share a Formal Contract

Units of automation logic that are classified as services must provide a contract in which the terms of engagement are defined.This principle is primarily concerned with service description documents that comprise the technical service contract and provide published details about the service, such as its programmatic interface, communication requirements, constraints, properties, usage policies, and even preferences.

Service Are Composable

The principle of composability applies to individual services, and strongly encourages that services be designed in support of aggregated assembly as composition controllers, members, or both.By ensuring that services are capable of participating in multiple compositions, an inventory of adaptive services can be accumulated.

Service Are Composable

The principle of composability applies to individual services, and strongly encourages that services be designed in support of aggregated assembly as composition controllers, members, or both.By ensuring that services are capable of participating in multiple compositions, an inventory of adaptive services can be accumulated.

Services Are Autonomous

Autonomy represents the governance a service has at the time of execution over the underlying application logic required to carry out the functions exposed by the service contract. The extent to which a service can control its underlying logic dictates the level of its autonomy

Services Are Stateless

Maintaining a condition of statelessness therefore benefits a service by increasing its scalability and availability. Furthermore, the processing of state information typically requires automation logic that is specific to the business task being executed. For services to maximize reuse potential, their context and underlying logic must be as generic (task-neutral) as possible

1.2.4 Service-oriented architectures have the following key characteristics:

- SOA services have self-describing interfaces in platform-independent XML documents. Web Services Description Language (WSDL) is the standard used to describe the services.
- SOA services communicate with messages formally defined via XML Schema (also called XSD). Communication among consumers and providers or services typically happens in heterogeneous environments, with little or no knowledge about the provider. Messages between services can be viewed as key business documents processed in an enterprise.
- SOA services are maintained in the enterprise by a registry that acts as a directory listing. Applications can look up the services in the registry and invoke the service. Universal Description, Definition, and Integration (UDDI) is the standard used for service registry.
- Each SOA service has a quality of service (QoS) associated with it. Some of the key QoS elements are security requirements, such as authentication and authorization, reliable messaging, and policies regarding who can invoke services.

1.2.5 WEB SERVICES

A web service [7] is a collection of protocols and standards used for exchanging data between applications or systems. "Web services are a new breed of Web application. They are self-contained,self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions, which can be anything from simple requests to complicated business processes.once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.

1.2.5.1 Advantages of Web services:

- WebServices communicate of over http protocol and can easily cross network boundaries.
- Web services support only the data types defined in the XSD type system, limiting the number of objects that can be serialized. This makes them interoperable across platforms, and are ideal for heterogeneous environments.

- Provides extensibility by allowing us to intercept the SOAP messages during the serialization and deserialization stages.
- Easy-to-create and deploy.

1.2.5.2 Characteristics

- A Web Service is accessible over the Web.
- Web Services communicate using platform-independent and language-neutral Web protocols.
- A Web Service shares schemas and contracts/interface that can be called from another program.
- A Web Service is registered and can be located through a Web Service Registry.
- Web Services support loosely coupled connections between systems.

1.2.5.3 Technologies

- **XML** (eXtensible Markup Language) markup language that underlies most of the specifications used for Web services.
- **SOAP** (Simple Object Access Protocol) is a network, transport, and programming language and platform neutral protocol that allows a client to call a remote service. The message format is XML.
- **WSDL** (Web services description language) An XML-based interface and implementation description language. The service provider uses a WSDL document in order to specify the operations a Web service provides.
- **UDDI** (universal description, discovery, and integration) Both a client-side API and a SOAP-based server implementation that can be used to store and retrieve information on service providers and Web services.

1.2.5.4 Web Services Design Principles

- Web-based Protocols: Web-services based on HTTP, protocols can traverse firewalls, can work in a heterogeneous environment
- Interoperability:– SOAP defines a common standard that allows different systems to interoperate

- Modularity:Service Components are useful in themselves, reusable, composable
- Availability:Services are available to systems that wish to use them, Services must be exposed outside of the particular system they are available in Machine-readable description used to identify the interface, the location and access information
- Implementation-independenc: Service interface available independent of the ultimate

  Implementation Published Searchable service repositories of service descriptions

1.2.5.5XML

XML [8] provides a standards-based method for describing data. XML is used extensively in building and consuming Web services. XML has the ability to describe data that is highly interoperable among many different systems on the Internet. Using the basic elements of XML we can define simple and complex data types and relationships. XML promotes the ability of Web services to communicate their data efficiently and effectively. It's this XML that ensures a consistent and accurate interpretation of the data when the service and consumer reside on different platforms.

- All the technologies in Web Services are XML based:Messaging,Description,Registry
- All are in XML

Why XML

XML is pure text with no binary data

- Applications read the XML
- Applications share data using XML .Any application can talk to any other application using XML (unlike binary) irrespective of the platform
- XML is a method for putting structured data in a text file

XML Rules

- Well formed:
  - Syntax is correct (all tags opened and closed)
- Valid:
  - All the elements matches the definitions written in the schema

- DTD
    - Document Type Definitions – Validates XML data against it
- XML Schema
    - Alternative to DTD with added functionality. It supports other data types not supported by DTD
    - Predefined Simple Types (integers,booleans,dateTime…)
    - User-defined datatypes (Complex Types)
    - Validations Restrictions to types
    - XML schema itself is an XML document !
- XML Processing
    - Read the XML documents XML processors (Parsers)
    - SAX (based on events)
    - DOM (reads the xml document and loads it in memory)
    - Python implements this interfaces in a package PyXML

CHAPTER 2

LITERATURE REVIEW

2.1 FEDERATED DATABASE SYSTEM:

A **federated database system** is a type of meta-database management system (DBMS) which transparently integrates multiple autonomous database systems into a single **federated database**. The constituent databases are interconnected via a computer network, and may be geographically decentralized. Since the constituent database systems remain autonomous, a federated database system is a contrastable alternative to the task of merging together several disparate databases. A federated database, or **virtual database**, is the fully-integrated, logical composite of all constituent databases in a federated database system.

EXSISTING SYSTEM AND RELATED WORKS:

2.1.1 McLeod and Heimbigner[9] were among the first to define a federated database system, as one which "define[s] the architecture and interconnect[s] databases that minimize central authority yet support partial sharing and coordination among database systems".

2.1.2 Christian SOUTOU[10] proposed a methodology for developing a federated database system. This principle permits a use conjoined of several databases without joining any global schema. The bottom-up approach is based on the homogenization and the integration of pie-existing and heterogeneous database schemas. Autonomous component database systems participate in a federation by sharing their data. It provides global integrated views called federated schemas.

Challenges:

- ✓ Standardized and simplified access to data
- ✓ Users can easily create integrated views of data and libraries of data store,that can be used multiple times.
- ✓ Data are provided in real time, from original sources, not from cumulative databases or duplicates.

Methodology:

- ✓ Ihe first to translates the local schemas of a component databases into the common data model
- ✓ The second step defines the export schemas from shared structure of a component schema.
- ✓ The next step selects a related set of export schemas to be integrated and integrates them in a federated schema.
- ✓ The last step defines extemal schemas which are the vkws for each user or class of federation users.

Advantages:

- ✓ Data remains at source,Latest data available,
- ✓ Little or no additional storage,Drill-down capabilities,
- ✓ Assuming a denormalized front-end schema, no major schema transform,
- ✓ Good for standard applications and related data sources, e.g., Peoplesoft, SAP and Siebel.

Disadvantages:

- ✓ Unclean data;Limited query processing;
- ✓ Imposes a heavy query load on data source systems and adapters;
- ✓ Low query performance - total system as fast as the slowest individual database;Limited types of data sources;
- ✓ Limited to no row, column and data element security (unless data source supports it);Data sources aware of queries;

2.1.3 Zhao-Ping Yu and Csaba Egyhazy[11] developed a query optimizer for the Cyrano prototype Federated Database System developed at Virginia Tech. They used a bottom-up evaluation method commonly seen in deductive systems. In Cyrano, queries and stored information is represented in classes, as in object-oriented systems. Consequently, the optimizer evaluates a query by repeatedly cycling through all base classes, and the base classes of the base classes. The implementation of a Semi-Naive algorithm produced considerable improvements

2.1.4 François Barbançon, Daniel P. Miranker[12] proposed a federated systems that integrating data from multiple sources must cope with semantic heterogeneity by reasoning over both the data and meta-data of their sources. SchemaSQL is one of a number of related higher-order languages, which have been proposed for succinctly expressing integrated views over heterogeneous sources..

2.1.5 Amit P.Sheth,James A.Larson[13] proposed a data federation approach.In a large modern enterprise, it is almost inevitable that different portions of the organization will use different systems to produce, store and search their critical data. Database federation is one approach to information integration in which a relational database management system serves as middleware providing transparent access to a number of heterogeneous data sources. . In this paper, the author describe the virtues of database federation, introduce several styles of database federation, and outline the conditions under which each style of federation should be used.

2.2 DATA INTEGRATION

Data integration involves combining data residing in different sources and providing users with a unified view of these data. In management circles, people frequently refer to data integration as "Enterprise Information Integration" (EII).

Enterprise Information Integration (EII), is a process of information integration, using data abstraction to provide a single interface (known as uniform data access) for viewing all the data within an organization, and a single set of structures and naming conventions (known as uniform information representation) to represent this data; the goal of EII is to get a large set of heterogeneous data sources to appear to a user or system as a single, homogeneous data source.

EXSISTING SYSTEM AND RELATED WORKS:

2.2.1 Vincenzo Di Lecce Alberto Amato Marco Calabrese [14] proposed a medical information system.Every organization contain many branches which are located at various place.the user may use of own interested database to store their data.The problem here is to integrate the data from heterogeneous data source and produce a single unified view of this

Problem Definition:

- ICT in the processes and activities of the health care system is an essential step to improve the resource management and limit the risk factors due to the daily exchange of information among the various functional units.

- Currently the "average" italian hospital has a fragmented structure composed of "information islands" where the activity is almost entirely automated. Good examples of these "information islands" can be the various departments constituting the hospital.

- Typically this situation occurs when the components of a clinical information system are purchased in different time and from different vendors. An interesting approach for integrating these applications is the Health Level Seven (HL7) protocol.

Problem Overview:

- Often patients are treated in various hospitals. Hospitals,medical doctors (MD), and other healthcare centres around the world require the service to send and receive healthcare data, including patient information and various laboratory reports.
- In many cases, MDs spend much of their time in searching for, correlating and revising information manually instead of focusing on information interpretation.
- This occurs because information sources are scattered across medical records and often unavailable at the time of care, thus reducing the effectiveness of preventive measures in case of extreme events such as sudden spreading pandemic viruses.
- The main challenge is to integrate information and processes thus providing services to patients and MDs wherever they are.

Challenges:

- data redundancy
- legacy systems
- unstructured data
- security

2.2.2 Li Xu and David W. Embley [15] proposed BGLaV(both global and local as view approach).There are two main basic approaches to data integration: Global-as-View (GaV) and Local-as-View (LaV). However, both GaV and LaV have their limitations. In a GaV approach,changes in information sources or adding a new information source requires revisions of a global schema and mappings between the global schema and source schemas. In a LaV approach,automating query reformulation has exponential time complexity with respect to query and source schema definitions. To resolve these problems, we offer BGLaV as an alternative point of view that is neither GaV nor LaV.

2.2.4 M.T. Baldassarre, D. Caivano, G. Visaggio [16] proposed EII management system which is a support tool to iterative reengineering of data and applications, developed in SERLAB. EII_MS provides a unified view of enterprise data over the various data sources in spite of original data heterogeneity and redundancy.

2.2.5 Yinzhang GUO, Guoyou ZHANG, Lipin XIE,Yubin XU [17] proposed a approach of business component reuse in the integration of rapid reconfigurable enterprise information system. The model of function component is provided, which is based on analyzing and abstracting function commonness from different components in business component domain, the compiling parameters of function component denote a business component type,and the running parameters denote the difference of business components. At the same time, the syntax description of the function components is presented. The reuse of business component is improved in enterprise information system integration by designing function component.

2.2.6 Peng Gong, Chi Yang, Dongping Shi [18] proposed research of heterogeneous information processing and integration method in Enterprise Management Information Systems, XML-based Global-as-View that is XGAV for short will be put forward in this passage.XGAV adopts such technologies as XML Schema, XSLT to describe Global-as-View formally, integration of heterogeneous information will be implemented effectively by this way

2.2.7 Ya,Li, Hairui,Wang, Lin,Jin, Yuping,Liu [19] proposed a Multi-agent technology.Multi-agent technology has been regarded as one of the promising technologies agent technology such as autonomy, distributed collaboration, and intelligence naturally fit with the characteristics of enterprise information integration. Finally,combine the verdicts from different sections of this paper, propose a general framework that utilizes Multi-agent technology for enterprise information integration, and choose the most suitable development platform.

2.2.8 DavidD F. Barrero, Mar′ IA D. R-moreno [20] proposed a searchy approach for integrating data from heterogeneous data sources. Searchy is a agent-based platform which, through the utilization of heterogeneous semantic wrappers, integrates information from arbitrary sources and translates them into semantic terms. Its distributed nature and non intrusive operation enables it to operate in a B2B environments where several entities share their information systems.

2.2.9 X. Y. Li, Y. Qian proposed a web service based enterprise information integration model. The legacy system is wrapped into web service through multi-agent technology.Then enterprises can dynamically integrate all web services distributed in different domains with different integration goals. Due to adoption of wrapping technology for legacy system, original security and stability of legacy system are guaranteed in the integration platform. Finally, the validity and practicability of the integration technology are verified through the application in the establishment of computer information integration platform for TE procedure.

2.4 Xpath Query and Decomposition:

EXSISTING SYSTEM AND RELATED WORKS:

2.4.1 Sandro Daniel Camillo, Carlos Alberto Heuser and Ronaldo dos Santos Mello proposed a system that provides semantic integration of heterogeneous XML sources, the same information at a semantic level may have different representations in XML. The syntax of an XML query depends on the structure of the specific XML source. Therefore, in order to obtain the same query result, one must write a specific query for each XML source.. Author presents CXPath (Conceptual XPath), a language for querying XML sources at the conceptual level, as well as a translation mechanism that converts a CXPath query to an

2.4.2 Georg Gottlob, Christoph Koch, Reinhard Pichler proposed a Contemporary XPath query engines evaluate queries in time exponential in the sizes of input queries, a fact that has gone unnoticed for a long time. Recently, the first main memory evaluation algorithm for XPath 1.0 with polynomial time combined complexity, i.e., which runs in polynomial time both with respect to the size of the data and the queries.A new XPath processing algorithms with improved time and space efficiency.

2.4.3 Stefan Böttcher, Rita Steinmetz proposed a XPath query optimization that can benefit from a tester that checks whether the intersection of data fragments described by two XPath expressions is empty for all valid database states. In this paper, we contribute a fast but incomplete intersection test for XPath expressions that reflects type constraints like sub-types and extensions defined in XML Schema.

2.4.4 Sven Groppe, Stefan Böttcher, Jinghua Groppe proposed a rule set which logically simplifies XPath queries by using a heuristic method in order to improve the processing time.Furthermore, we show how to substitute the XPath 2.0 intersect and except operators in a given XPath query with computed filter expressions.

2.4.5 Le Thi Thu Thuy, Doan Dai Duong , Virendrakumar C. Bhavsar1 and Harold Boley proposed a system In order to access data from various different data repositories, in Global-As-View approaches an input query is decomposed into several subqueries.

- ✓ Normally, this decomposition is based on a set of mappings, which describe the correspondence of data elements between a global schema and local ones. However, building mappings is a difficult task,especially when the number of participating local schemas is large.
- ✓ In our approach, an input query is automatically decomposed into subqueries without using mappings.
- ✓ An algorithm is proposed to transform a global path expression (e.g., an XPath query) into local path expressions (e.g., XPath queries) executable in local schemas. This algorithm transforms parts of a path expression from right to left. This transformation is applied from the bottom to the top of a tree and depends on structures of local

.4.6 Felipe Victolla Silveira and Carlos A. Heuser proposed a approach ,here the problem of integration of XML data sources is decomposed in two problems:

(1) that of fragmentation of a global graph-like model (e.g.,an ER model) into several local graph-like models conceptually representing data sources and

(2) that of mapping the local graph-like model into an XML tree-like schema. This paper presents a set of fragmentation operators specifically designed for our approach, as well as a query decomposition mechanism that allows a query stated at the conceptual level to be decomposed into an XQuery statement at the XML level.

2.4.7 Radha Senthilkumar, G. B. Rakesh, N.Sasikala, M.Gowrishankar, A. Kannan: The XPath language is based on a tree representation of the XML document.Here an optimization plan is proposed, to unnest and optimize nested XPath query for XML Inter and Intra document relationship. We propose an enhanced variant of kappa join which is used for query unnesting.. Unnesting is performed by means of enhanced variant of kappa join taking into account the XML relationship. Unnested query is converted to a internal PAT(Pattern) representation. This PAT expression is optimized by deterministic transformation on queries using the structure knowledge of XML data and structure-related semantics.

CHAPTER 3

SYSTEM ANALYSIS

3.1 DRAWBACKS OF EXISTING SYSTEM:

3.1.1 EAI

Enterprise Application Integration

- ✓ Complex to maintain
- ✓ Not highly scalable
- ✓ Complex to destroy
- ✓ Expensive to setup

3.1.2 ETL

Extraction Transformation Loading

- ✓ Database access as opposed to logs
- ✓ Batch processing vs real time
- ✓ Require extensive re-work to handle real time data
- ✓ Rigid-DBMS,application packages,flat files.
- ✓ Back office,un-attended integration

3.1.3 DTC

Detect Transform Communicate

- ✓ Doesn't handle message queues
- ✓ Doesn't handle proprietory functions
- ✓ Doesn't validate transaction process

The following are some of the available tools which can be used to integrate data from heterogeneous data source:

- ✓ Pentaho
- ✓ Informatica
- ✓ XAware
- ✓ Allora and DBMoto

3.2 PROPOSED SYSTEM:

In fast running business world its imposiible to store all information in a single place as well as in single system and platform independent.Every organisation containing many branches which are locate at distinct places. For effective decision making we need all the information which are located at different places at right time in right format.

In proposed approach ,data integration takes place on

- ✓ On-demand basis
- ✓ Also getting real time data

The data in various source can be consider as a web service and can be integrated using service oriented architecture.if we using web service its easy to integrate because its platform independent,can be called using internet etc.Here querying the database,accessing the database,query decomposition ,result integration everything are considered as web service.It is published in local host using Internet Information Service(IIS)

Hospital Information Sytem:

Problem Overview:

- ➢ In today's extremely challenging business environment, every organisation **measuring their success** by comparing their performance over many years.
- ➢ We need to integrate all departments individual performance together for comparison

Challenge:

- Integrating all these data is **highly valuable** but **technically challenging**.
- Integrating the data from disperate sources in different platform

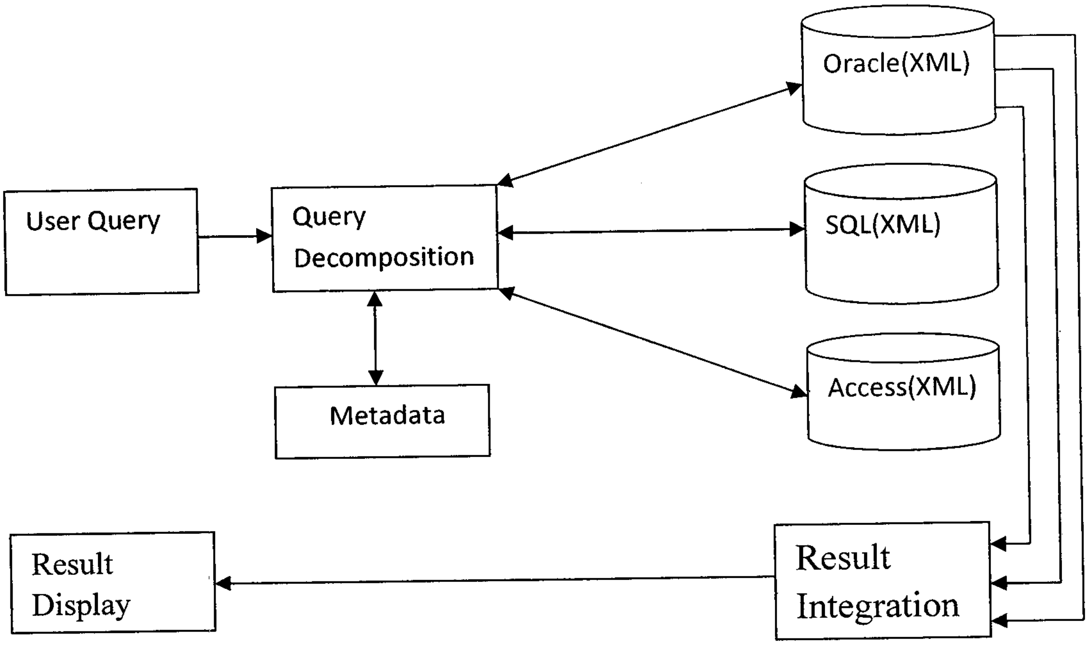


Fig 4.Project Architecture

End Users:

- Analysis

  --Analyser comparing their results

  --Goal

- Reporting

--Reporting their work to higher officials for further improvement of organisation

Expected Advantage of this Proposed System:

- ✓ Space complexity reduced.
- ✓ Real time data is retrieved

CHAPTER 4

SYSTEM SPECIFICATION

4.1 HARDWARE REQUIREMENTS

Processor	: Pentium III and above
Clock speed	: 550MHz
Hard Disk	: 20GB
RAM	: 128MB or above
Cache Memory	: 512KB
Operating System	: any Windows version
Monitor	: Color Monitor
Keyboard	: 104Keys
Mouse	: 3Buttons

4.2 SOFTWARE REQUIREMENTS

Database	: MS Access, SQL ,XML
Database drivers	: ODBC driver,SQL-JDBC driver
Other tools(optional)	: Altova(Convert any database into XML)

4.3 SOFTWARE DESCRIPTION:

Visual Studio:

- ✓ Complete development environment with comprehensive application development features, including improved visual designers, code editors, and programming languages for optimum productivity.
- ✓ Develop and debug multi-tier server applications from with the unified development enviroment - even remotely.
- ✓ Build tools that extend the Visual Studio Integrated Development enviroment using the Visual Studio Software Developers Kit.

Microsoft Visual Studio Professional 2005 is one of the most significant developer tools available for building high-performance, multi-tier applications for Windows, the Web, and mobile devices. Designed for developers working alone or in small teams, Visual Studio Pro 2005 features integrated visual database tools for designing databases, tables, and stored procedures. You can also design, debug, and deploy multi-tier applications; and build a wide array of high-performance solutions faster than ever.With Visual Studio Pro 2005's powerful, enterprise-class application platform, you can also easily create and application platform, you can also easily create and deploy client applications, and automatically publish and maintain applications and their dependencies with integrated Click Once support. With more than 50 new controls and hundreds of built-in services for site security, personalization and look and feel, Visual Studio Pro 2005 gives you the tools it takes to build fast, interactive Web applications.

VB.NET:

- ✓ Visual Basic . Net has flexibility , allowing one or more language to interoperate to provide the solution. This Cross Language Compatibility allows to do project at faster rate.
- ✓ Visual Basic . Net has Common Language Runtime , that allows all the component to converge into one intermediate format and then can interact.
- ✓ Visual Basic . Net has provide excellent security when your application is

- ✓ Visual Basic .Net has flexibility, allowing us to configure the working environment to best suit our individual style. We can choose between a single and multiple document interfaces, and we can adjust the size and positioning of the various IDE elements.
- ✓ Visual Basic . Net has Intelligence feature that make the coding easy and also Dynamic help provides very less coding time.
- ✓ The Visual Basic .Net language is quite powerful – if we can imagine a programming task and accomplished using Visual Basic .Net.
- ✓ After creating a Visual Basic . Net application, if we want to distribute it to others we can freely distribute any application to anyone who uses Microsoft windows. We can distribute our applications on disk, on CDs, across networks, or over an intranet or the internet.
- ✓ The working environment in Visual Basic .Net is often referred to as Integrated Development Environment because it integrates many different functions such as design, editing, compiling and debugging within a common environment. In most traditional development tools, each of separate program, each with its own interface.

SQL Server 2005

SQL is a relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. It is a popular choice of database for use in web applications, and can be used as central component for open source project development.

ODBC (Open database Connectivity)driver

Open Database Connectivity (ODBC) is Microsoft's strategic interface for accessing data in a heterogeneous environment of relational and non- relational database management systems. Based on the Call Level Interface specification of the SQL Access Group, ODBC provides an open, vendor- neutral way of accessing data stored in a variety of proprietary personal computer, minicomputer, and mainframe databases.

CHAPTER 5

PROJECT DESCRIPTION

5.1 PROBLEM DEFINITION:

A hospital may contain many branches which are locatd at different places which may use of own interested database to store the hospital information. The data available is huge and not in unique format.It is difficult for a user to manually analyze these data and make decisions. The problem here is to extract the heterogeneous data source, tranform them into a consistent format and produce a single unified view of this data.Users are totally unaware of the internal structure of the data and also how to write queries to view the internal data.A system must be designed which provides a user interactive interface to integrate the data. An effective federated database management must be designed for hospital information system to integrate the data from heterogeneous database on on-demand basis.

5.2 PROJECT OVERVIEW:

In this project,integrating the data from heterogeneous data source takes place ,where web services are placed in local system and web services are placed in remote server and comparing both. The source for the project is from a hospital database. The entire project is divided in to two phases as follows:-

- **Phase I:** Web services are placed in local system.
- **Phase II:** Web services are placed in remote server.

5.3 MODULES:

Enterprise Information Integration(EII)

The system is composed of a number of sub-systems, which correspond to each stage of data integration. These stages are A) User login-user can login the system,the user may be of any type like administrator,higher officials etc, B) Querying the database-user can query the database based on their requirement in the format of XML,C) Query decomposition-It split the query into several sub-query based on local and global schema.D) Database integration the data from different database. F) Result display The overall diagram of enterprise information integration(eii) is shown in Fig.3.1

5.3.1. User Login: User can log in to the system to query the database.The user must be the authorised one,for checking whether the person is authorised ,username and password is checking,if its correct ,the user can query the database,suppose if its wrong it will error message like "username or password incorrect,please enter correct username and password.

5.3.2 Querying the database:

Queries are one of the things that make databases so powerful. A "query" refers to the action of retrieving data from your database. Usually, you will be selective with how much data you want returned. If you have a lot of data in your database, you probably don't want to see everything. More likely, you'll only want to see data that fits a certain criteria.For example, you might only want to see how many patients in your database live in a given city. Or you might only want to see which individuals have registered with your database within a given time period. Database queries consist of patterns which match either individual segments, sequences of segments or parts of the hierarchical description of an utterance. The result of a query is a segment list which contains the labels and start and end times of each segment or sequence of segments which matched the query, along with the utterances to which they belong.

Queries can relate to any level of the hierarchy defined by the template file.The type of segments returned depends on the form of the query.There are two basic kinds of query: simple queries which relate to single segments, and compound queries which relate to either sequences of segments or to segments at different levels in the hierarchy.

The trend in data integration has favored loosening the coupling between data. This may involve providing a uniform query-interface over a mediated schema , thus transforming a query into specialized queries over the original databases. One can also term this process **"view-based query-answering"** because each of the data sources functions as a view over the (nonexistent) mediated schema.

5.3.3 Global-as-view and Local-as-view

There are two major initiatives to integrate data and answer queries without materializing a query processing algorithms proposed in the literature for LAV, and describes the principle GAV data integration systems and the form of query processing they adopt. In a GAV approach, query reformulation reduces to simple rule unfolding standard execution of views in ordinary databases).However, changes in information sources or adding a new information source requires a database administrator (DBA) to revise the global schema and the mappings between the global schema and source schemas. Thus, GAV is not scalable for large applications. LAV scales better, and is easier to maintain than GAV because DBAs create a global schema independently of source schemas. Then, for a new (or changed) source schema, the DBA only has to give (adjust) a *source description* that describes source relations as views of the global schema. Automating query reformulation in LAV,however, has exponential time complexity with respect to query and source schema definitions. Thus, LAV has low query performance when users frequently pose complex queries.

Global-as-view:

The GAV approach is procedural as the mapping explicitly says how to retrieve data from the source to compute certain answers.Computing certain answers is easy for all monotone queries (simple unfolding of the query).However, in GAV it is not easy to extend the system with new sources (all views may have to be redefined).

3.2.1.2Local-as-view:

LAV approach is declarative as the content of the source is characterized by views over the global schema.It favors the extensibility of the system (a new source just needs a new view over the global schema).

Global Schema:

The Global Schema is the structure of the Content System database. It is a database that is separate from the document store databases, and controls connections and information to each of the document stores. The Global Schema Settings page allows the Administrator to modify the settings for database connection times and the size of the connection pool.

Local Schema:

Local schemas are argued to be important in verifying conceptual data requirements before

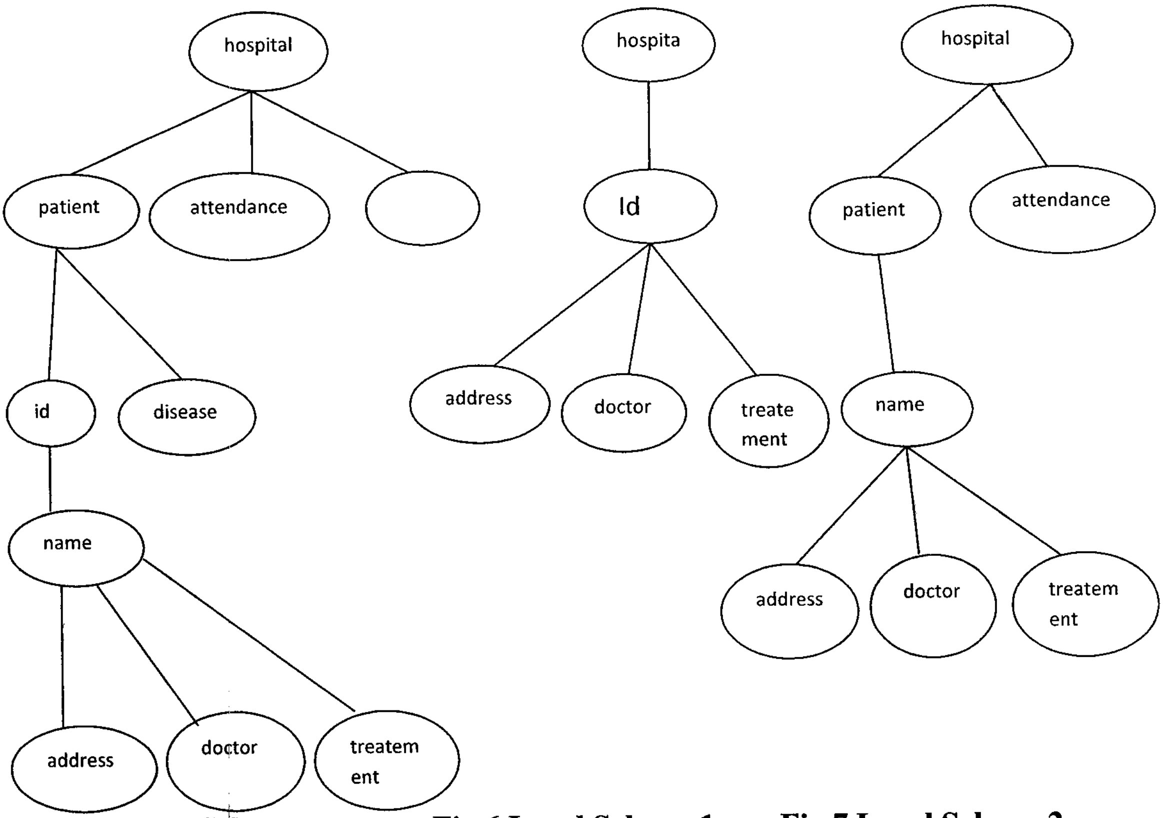


Fig 5 Global Schema **Fig 6 Local Schema1** **Fig 7 Local Schema2**

5.3.4 Xpath Query:

XPath, the **XML Path Language,** is a query language for selecting nodes from an XML document. In addition, XPath may be used to compute values (e.g., strings, numbers, or Boolean values) from the content of an XML document. The XPath language is based on a tree representation of the XML document, and provides the ability to navigate around the tree, selecting nodes by a variety of criteria. In popular use (though not in the official specification), an XPath expression is often referred to simply as an XPath.

5.4 Query Decomposition:

Query decomposition service accepts xpath query built based on global schema and outputs the sub-query that conforms to the local schema for which the sub query need to be generated. The xpath query based on global schema of the form $/p_1/p_2/p_3/\ldots/p_{n-1}/p_n$ is evaluated from right to left as in xpath query. It is the actual result which the user wants to get from federated system .The local schema tree is traversed from leaf to root level. If p_n does not exist in local schema, it can be concluded that there is no sub-query for this schema.

p_i for (i = (n-1)…1) will be performed from the ancestor nodes of the matched node. Instead of searching the whole tree only the ancestor nodes of last matched node need to be searched. This can significantly reduce the search time. If node p_i is found, then p_i is concatenated with p_n to form query.

The algorithm for query decomposition is shown below

Algorithm:

Input : userquery Q_{global} based on global schema, local schema S

Output: sub-query for S

```
Service splitter(userquery, local schema)
1. Initialize the variables 'query' and 'subquery' to NULL.
2. Initialize the arrays parts[], condition[] and value[] to NULL.
3. Split Q_global (global schema) and store it in an array parts[].
4. For each parts[i] repeat
       If parts[i] has condition and value then
               i.  Split them and store in condition[i] and value[i] arrays respectively.
               ii. Create subquery based on local schema with condition
       Else
               i.  Create subquery based on local schema
5. Return subquery.
```

Data access service accepts the sub-query from query decomposition service. This xpath sub-query conforms to local schema of data source in which data access service runs as opposed to user query based on global schema before decomposition.

This service executes the xpath query and outputs a dataset that contains the requested data found in this data source. Each data source has an instance of data access service running at a port receiving the xpath sub-query, executes the query and provides the dataset.

The algorithm for sub query execution and data set generation is as follows

Algorithm:

Input : xpath subquery

Output :Dataset containing requested data from respective XML data source

```
Service Dataaccess (subquery)
6. Create a navigator XNav for XML Document
7. Evaluate subquery using XNav
```

```
While there exist nodes in XNodeIter
If node has attributes
While there exist node.attribute
add the value to the Dataset
End While
If node has child elements
While node.child elements
add the value to the Dataset
End While
return Dataset
```

Result Integrator Service accepts the dataset from different data access service containing the data from multiple data sources. It combines the datasets eliminating any duplicates, orders the data for consistent display of records and sorts the records if required. The combined dataset is displayed to the user.

Algorithm:

```
Input    : Dataset DSA,DSB....DSn
Output   : Merged Dataset DSM
Service ResultIntegrator (DSA,DSB....DSn)
Dataset DSM := null
DSM :=Merge(DSA,DSB....DSn )
DSM :=Remove_duplicate_records(DSM)
DSM :=Sort(DSM,sortcolumnname,ASC/DESC)
return DSM
```

3.5 Database Conversion:

- ✓ First source database are converted into XML.
  - o Access database are converted into XML
  - o SQL database are converted into XML
  - o Oracle database are converted into XML

    First converting all database into XML.A common data model is needed to integrate the data in efficient manner.If all database in same data model,user query retrieve data in quick time.

.6 Result Integration:

Data Integration provides a unified view of the business data that is scattered hroughout an organization. Data Integration facilitates query over autonomous and ieterogeneous data sources through a common and uniform schema (global schema).User giving Xpath query as input.Query splitted and given to various database by checking global and local schema.Query execution takes place in source XML database side.Finally results from different database are integrated and produce a single unified view of this data.

5.7 Result Display:

After user login,user query,query decomposition,query execution ,result integration,result display module takes place.

5.8 Database Technology used

For hospital information system,there are three different database are used to store user information.

- **Source Database**
  - Access
  - SQL
  - XML
  - Oracle
- **XML converted database:**

  All the database in XML format.

  Access->XML

  SQL- >XML

  Oracle ->XML

5.9 Database Technology used

- ✓ **Access**
- ✓ **SQL**
- ✓ **XML**

For federated database system , database used by local hospital is used. There are totally three tables, namely patient attendance and medical .There are totally 1500 records are

CHAPTER 6

RESULTS AND DISCUSSION

For experimental study a database containing 10,000 records is used. The databases which serve as data sources are geographically distributed and run in heterogeneous platform with structural and semantic heterogeneity. The data are used are pertinent to hospital organization which have number of branches. All these data sources are initially converted to XML databases to suite xpath query decomposition. The local schemas are used for source identification and federated schema generation. Initially 2000 records were populated. The web services were executed from local host. The data federation is performed for the given query. The above process was repeated by incrementing records by 1000 for every run, till the count reaches 10000 records. The time taken to execute same query increased proportionally [Figure 5]

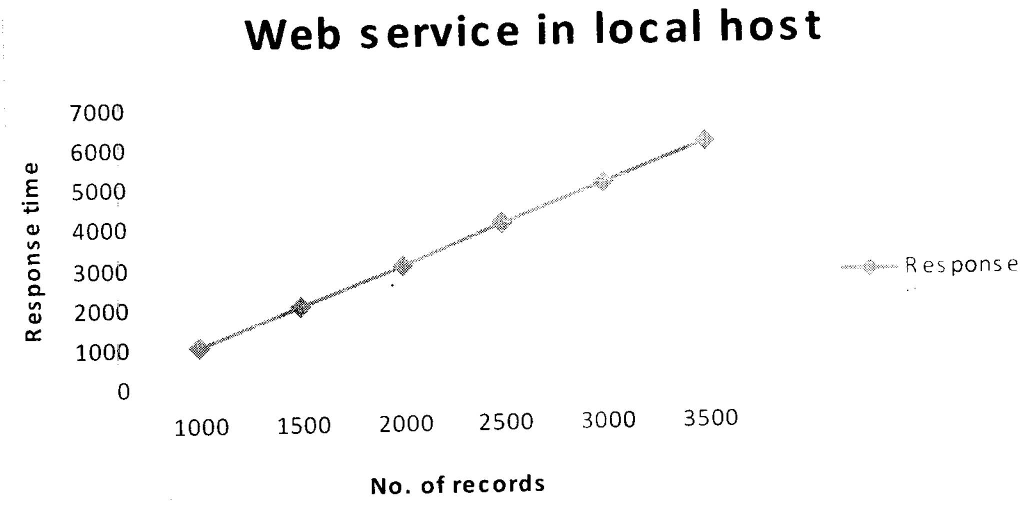
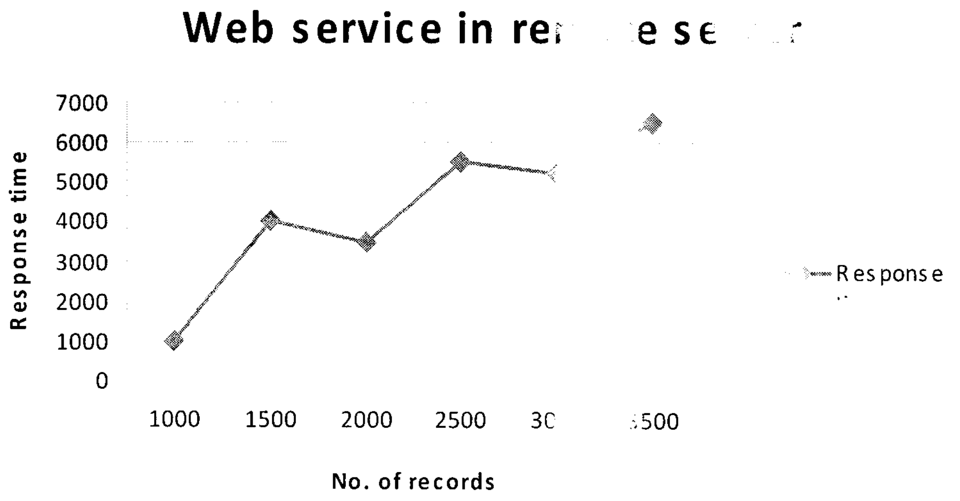


Figure 5 Response Time for datafederation (web services are located in local host)



igure 6 Response Time for datafederation (rvice cated remote server)

veb services were placed in a remote server the al rocess was repeated. The aken to execute query is not proportional wi nber o ds [Figure 6]. The factors erver load, bandwidth availability affects the ation t

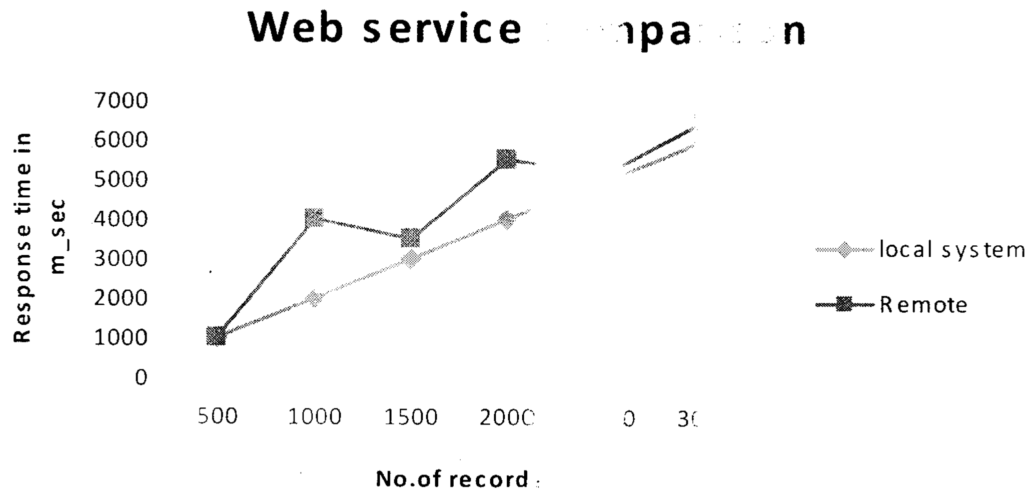


Fig 7.Web service c on

the comparison for placing the web servi local and remote system are lered and graphs are plot.

CHAPTER 7

CONCLUSION AND FUTURE OUTLOOK

Thus a new data integration system is proposed which integrate the data from heterogeneous database on on-demand basis and also getting real time data is achieved .Here we are not storing the data anywhere.Extracting the data from heterogeneous database ,transforming the data and loading the data into user console.Just creating a view not storing the data anywhere so time and space complexity is reduced.In proposed system source IP address is given directly before executing the program.Thus data integration technique like Enterprise Information Integration(EII) works better than Enterprise application Integration(EAI) and Extraction Transformation Loading(ETL) technique.In future integrating the data from heterogeneous database on on-demand basis is achieved by giving the different source IP address dynamically.

APPENDIX I

SOURCE CODE

Coding for Accesing Information in Local System:

a) Source code for adding patient details in Access:

```
Imports System.Data
Imports System.Data.OleDb

Public Class DataAccess
    Public conString As String = ""
    Public connection As New OleDbConnection
    Public command As New OleDbCommand
    Public adapter As New OleDbDataAdapter
    Public dTable As New DataTable
    Dim DataSRC As DataSet

    Public Sub New()
        Try
            ' conString = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Hospital;Data Source=."
            conString = "provider=microsoft.jet.oledb.4.0; data source=E:\Hospital1\hOSPITAL-
Acces\Access\Hospital.mdb "
            'conString = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Mobile_Processing;Data Source=."
            connectDB()
        Catch ex As Exception
            MsgBox("Exception  : " & ex.Message)
        End Try
    End Sub

    Sub connectDB()
        Try
            connection = New OleDbConnection(conString)
            'OleDbConnection..ClearPool(connection)
            If connection.State = ConnectionState.Open Then
                connection.Close()
                connection.Open()
            Else
                connection.Open()
            End If
        Catch ex As Exception
            MsgBox("Exception  : " & ex.Message)
        End Try
    End Sub
```

```
  Try
    connection.Close()
  Catch ex As Exception
    MsgBox("Exception  : " & ex.Message)
  End Try
End Sub

Sub ExecuteQuery(ByVal qryString As String)
  Try
    'connectDB()
    command = New OleDbCommand(qryString, connection)
    command.CommandTimeout = 100000
    command.ExecuteNonQuery()
    command.Dispose()
    'disConnectDB()
  Catch ex As Exception
    MsgBox("Exception  : " & ex.Message)
  End Try
End Sub
Sub executereader(ByVal qryString As String)
  Try
    'connectDB()
    command = New OleDbCommand(qryString, connection)
    command.CommandTimeout = 100000
    command.ExecuteReader(qryString)
    command.Dispose()
    'disConnectDB()
  Catch ex As Exception
    MsgBox("Exception  : " & ex.Message)
  End Try
End Sub

Public Function getDataTable(ByVal qrySel As String) As DataTable
  Try
    dTable = New DataTable
    adapter = New OleDbDataAdapter(qrySel, conString)
    adapter.Fill(dTable)
  Catch ex As Exception
    MsgBox("Exception  : " & ex.Message)
  End Try
  Return dTable
End Function
Public Function getDataset(ByVal qrySel As String) As DataSet
  Try

    Dim AdaSRC As OleDbDataAdapter
    DataSRC = New DataSet
    AdaSRC = New OleDbDataAdapter(qrySel, conString)
    AdaSRC.Fill(DataSRC)
```

```
          MsgBox(ex.Message.ToString)
      End Try
      Return DataSRC
   End Function


   Sub clrText(ByVal obj As Control)
      Try
          Dim rec As Control = Nothing
          For Each rec In obj.Controls
             If TypeOf rec Is TextBox Then
                'rec.Text = " "
             End If
          Next
      Catch ex As Exception
          ' MessageBox.Show("Exception  : " & ex.Message)
      End Try
   End Sub


   Protected Overrides Sub Finalize()
      MyBase.Finalize()
   End Sub
End Class
```

b) Source code for adding patient details in SQL:

```
Imports System.Data
Imports System.Data.SqlClient

Public Class DataAccess
   Public conString As String = ""
   Public connection As New SqlConnection
   Public command As New SqlCommand
   Public adapter As New SqlDataAdapter
   Public dTable As New DataTable
   Dim DataSRC As DataSet

   Public Sub New()
      Try
          conString = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Hospital;Data Source=."
          '
          'conString = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Mobile_Processing;Data Source=."
          connectDB()
      Catch ex As Exception
```

```
    End Try
End Sub

Sub connectDB()
    Try
        connection = New SqlConnection(conString)
        SqlConnection.ClearPool(connection)
        If connection.State = ConnectionState.Open Then
            connection.Close()
            connection.Open()
        Else
            connection.Open()
        End If
    Catch ex As Exception
        'MessageBox.Show("Exception  : " & ex.Message)
    End Try
End Sub

Sub disConnectDB()
    Try
        connection.Close()
    Catch ex As Exception
        ' MessageBox.Show("Exception  : " & ex.Message)
    End Try
End Sub

Sub ExecuteQuery(ByVal qryString As String)
    Try
        'connectDB()
        command = New SqlCommand(qryString, connection)
        command.CommandTimeout = 100000
        command.ExecuteNonQuery()
        command.Dispose()
        'disConnectDB()
    Catch ex As Exception
        ' MessageBox.Show("Exception  : " & ex.Message)
    End Try
End Sub
Sub executereader(ByVal qryString As String)
    Try
        'connectDB()
        command = New SqlCommand(qryString, connection)
        command.CommandTimeout = 100000
        command.EndExecuteReader(qryString)
        command.Dispose()
        'disConnectDB()
    Catch ex As Exception
        'MessageBox.Show("Exception  : " & ex.Message)
    End Try
```

```
 Public Function getDataTable(ByVal qrySel As String) As DataTable
   Try
     dTable = New DataTable
     adapter = New SqlDataAdapter(qrySel, conString)
     adapter.Fill(dTable)
   Catch ex As Exception
     ' MessageBox.Show("Exception  : " & ex.Message)
   End Try
   Return dTable
 End Function
 Public Function getDataset(ByVal qrySel As String) As DataSet
   Try

     Dim AdaSRC As SqlDataAdapter
     DataSRC = New DataSet
     AdaSRC = New SqlDataAdapter(qrySel, conString)
     AdaSRC.Fill(DataSRC)
   Catch ex As Exception
     ' MessageBox.Show(ex.Message.ToString)
   End Try
   Return DataSRC
 End Function



 Sub clrText(ByVal obj As Control)
   Try
     Dim rec As Control = Nothing
     For Each rec In obj.Controls
       If TypeOf rec Is TextBox Then
         'rec.Text = " "
       End If
     Next
   Catch ex As Exception
     ' MessageBox.Show("Exception  : " & ex.Message)
   End Try
 End Sub
 Protected Overrides Sub Finalize()
   MyBase.Finalize()
 End Sub
End Class
```

Attendance Details:

```
Imports System.Data
Partial Class Attentance
```

```
  Dim obj As New DataAccess
  Dim objx As New BusinessLogic

  Protected Sub btnsubmit_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnsubmit.Click
      Try
        If TextBox2.Text.Trim <> "" Then
          Dim ss As String = JD1.SelectedValue.ToString
          Dim s1() As String = ss.Split(" ")
          objx.CreateProduct("insert into Attendance values(" & Val(txtuserid.Text) & ",'" &
TextBox1.Text.Trim & "','" & DropDownList1.SelectedItem.ToString.Trim & "','" &
DropDownList2.SelectedItem.ToString.Trim & "','" & txtphone.Text.Trim & "','" &
s1.GetValue(0).ToString & "','" & DropDownList3.SelectedItem.ToString.Trim & "','" &
TextBox2.Text.Trim & "')")
          Dim s As String = "Attendance"
          xml(s)
          clr()
        End If
      Catch ex As Exception
        MsgBox(ex.Message)
      End Try
    End Sub
    Sub clr()
      txtuserid.Text = ""
      TextBox1.Text = ""
      txtphone.Text = ""
      TextBox2.Text = ""
    End Sub
    Sub xml(ByVal s As String)
      Try
        Dim qry As String
        qry = "select * from  " & s
        obj.connectDB()
        Dim DS1 As New DataSet
        DS1 = obj.getDataset(qry)
        obj.disConnectDB()
        Dim str As String = Server.MapPath("~\xml\")
        DS1.WriteXml(str & s & ".xml")
      Catch ex As Exception
        MsgBox(ex.Message)
      End Try
    End Sub

    Protected Sub txtuserid_TextChanged(ByVal sender As Object, ByVal e As
 System.EventArgs) Handles txtuserid.TextChanged

    End Sub
    Sub cnt()
      Try
```

```
            qry = "select pname from patient where uid=" & Val(txtuserid.Text) & ""
            obj.connectDB()
            Dim ds As New DataSet
            ds = obj.getDataset(qry)
            obj.disConnectDB()
            If ds.Tables(0).Rows.Count - 1 <= 0 Then
                TextBox1.Text = ds.Tables(0).Rows(0).Item(0).ToString
            Else
                TextBox2.Text = ds.Tables(0).Rows(0).Item(0).ToString
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Protected Sub DropDownList3_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles DropDownList3.SelectedIndexChanged
        Try
            TextBox2.Text = ""
            If DropDownList3.SelectedItem.ToString.Trim = "None" Then
                TextBox2.Text = "None"
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub

    Protected Sub DropDownList2_SelectedIndexChanged(ByVal sender As Object, ByVal e
As System.EventArgs) Handles DropDownList2.SelectedIndexChanged
        Try
            txtphone.Enabled = True
            If DropDownList2.SelectedItem.ToString.Trim = "Present" Then
                txtphone.Enabled = False
            End If
        Catch ex As Exception
            MsgBox(ex.Message)
        End Try
    End Sub
End Class
```

Adding Patient Details using Front End:

```
Imports System.Data
Partial Class PatientEntry
    Inherits System.Web.UI.Page
    Dim objx As New BusinessLogic
    Dim qry As String
```

```
  Dim bo As Boolean = False


  Protected Sub Button1_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles btnsubmit.Click
      Try
        If txtName.Text.Trim <> "" Then


            objx.CreateProduct("insert into Patient values(" & Val(txtuserid.Text) & ",'" &
txtName.Text.Trim & "','" & txtaddress.Text.Trim & "','" & txtphone.Text.Trim & "','" &
DropDownList3.SelectedItem.ToString.Trim & "','" & txtPassword.Text.Trim & "','" &
UName.Text.Trim & "','" & DropDownList2.SelectedItem.ToString.Trim & "','" &
JD1.SelectedValue.ToString & "','" & TextBox1.Text.Trim & "','" & TextBox2.Text.Trim &
"')")
            Dim s As String = "patient"
            xml(s)
            cnt()
            MsgBox("Patient Name SuccessFully Added")
            clr()

        End If
      Catch ex As Exception
        MsgBox(ex.Message)
      End Try
    End Sub
    Sub clr()
      txtName.Text = ""
      txtaddress.Text = ""
      txtPassword.Text = ""
      UName.Text = ""
      txtphone.Text = ""
      TextBox1.Text = ""
      TextBox2.Text = ""
    End Sub
    Sub xml(ByVal s As String)
      Try
        Dim qry As String
        qry = "select * from  " & s
        obj.connectDB()
        Dim DS1 As New DataSet
        DS1 = obj.getDataset(qry)
        obj.disConnectDB()
        Dim str As String = Server.MapPath("~\xml\")
        DS1.WriteXml(str & s & ".xml")
      Catch ex As Exception
        MsgBox(ex.Message)
      End Try
```

```
 Sub cnt()
    Try
       qry = "select * from Patient"
       obj.connectDB()
       Dim ds As New DataSet
       ds = obj.getDataset(qry)
       obj.disConnectDB()
       'If ds.Tables(0).Rows.Count - 1 <= 0 Then
       txtuserid.Text = ds.Tables(0).Rows.Count + 1
       ' Else
       ' txtuserid.Text = CInt(ds.Tables(0).Rows(0).Item(0).ToString) + 1
       'End If
     Catch ex As Exception
        MsgBox(ex.Message)
     End Try
  End Sub

  Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs)
Handles Me.Load
      cnt()

  End Sub
End Class
```

c) Source code for adding patient details in XML:

```
Imports System.Data
Imports System.Data.OleDb

Public Class DataAccess
   Public conString As String = ""
   Public connection As New OleDbConnection
   Public command As New OleDbCommand
   Public adapter As New OleDbDataAdapter
   Public dTable As New DataTable
   Dim DataSRC As DataSet

   Public Sub New()
      Try
        ' conString = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Hospital;Data Source=."
         conString = "provider=microsoft.jet.oledb.4.0; data source=G:\Hospital1\Hospital-
XML\Access\Hospital.mdb "
         'conString = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Mobile Processing;Data Source=."
```

```
    Catch ex As Exception
        MsgBox("Exception  : " & ex.Message)
    End Try
End Sub

Sub connectDB()
    Try
        connection = New OleDbConnection(conString)
        'OleDbConnection..ClearPool(connection)
        If connection.State = ConnectionState.Open Then
            connection.Close()
            connection.Open()
        Else
            connection.Open()
        End If
    Catch ex As Exception
        MsgBox("Exception  : " & ex.Message)
    End Try
End Sub

Sub disConnectDB()
    Try
        connection.Close()
    Catch ex As Exception
        MsgBox("Exception  : " & ex.Message)
    End Try
End Sub

Sub ExecuteQuery(ByVal qryString As String)
    Try
        'connectDB()
        command = New OleDbCommand(qryString, connection)
        command.CommandTimeout = 100000
        command.ExecuteNonQuery()
        command.Dispose()
        'disConnectDB()
    Catch ex As Exception
        MsgBox("Exception  : " & ex.Message)
    End Try
End Sub
Sub executereader(ByVal qryString As String)
    Try
        'connectDB()
        command = New OleDbCommand(qryString, connection)
        command.CommandTimeout = 100000
        command.ExecuteReader(qryString)
        command.Dispose()
        'disConnectDB()
    Catch ex As Exception
    End Try
  End Sub

  Public Function getDataTable(ByVal qrySel As String) As DataTable
    Try
      dTable = New DataTable
      adapter = New OleDbDataAdapter(qrySel, conString)
      adapter.Fill(dTable)
    Catch ex As Exception
      MsgBox("Exception  : " & ex.Message)
    End Try
    Return dTable
  End Function
  Public Function getDataset(ByVal qrySel As String) As DataSet
    Try

      Dim AdaSRC As OleDbDataAdapter
      DataSRC = New DataSet
      AdaSRC = New OleDbDataAdapter(qrySel, conString)
      AdaSRC.Fill(DataSRC)
    Catch ex As Exception
      MsgBox(ex.Message.ToString)
    End Try
    Return DataSRC
  End Function

  Sub clrText(ByVal obj As Control)
    Try
      Dim rec As Control = Nothing
      For Each rec In obj.Controls
        If TypeOf rec Is TextBox Then
          'rec.Text = " "
        End If
      Next
    Catch ex As Exception
      ' MessageBox.Show("Exception  : " & ex.Message)
    End Try
  End Sub


  Protected Overrides Sub Finalize()
    MyBase.Finalize()
  End Sub
End Class
```

Source code for accessing details in Remote Server:

a) Access:

```
Imports System.Web
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Data
Imports System.Data.SqlClient
<WebService(Namespace:="http://tempuri.org/")> _
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Public Class Service
    Inherits System.Web.Services.WebService


    Public sqlConStr As String = "Integrated Security=SSPI;Persist Security Info=False;Initial
Catalog=Hospital;Data Source=."

    Public SQL_SERVER As SqlConnection = Nothing
    Dim sqlCommand As SqlCommand
    Dim dSet As DataSet = Nothing
    Dim AdaSRC As SqlDataAdapter
    Public pat As String
    Public displayQuery As String = ""

    <WebMethod()> Public Sub Connect()
        Try
            SQL_SERVER = New SqlConnection(sqlConStr)
            If SQL_SERVER.State = ConnectionState.Open Then
                SQL_SERVER.Close()
                SQL_SERVER.Open()
            Else
                SQL_SERVER.Open()
            End If
        Catch ex As Exception
            MsgBox(ex.Message.ToString, , "Connect()")
        End Try
    End Sub

    <WebMethod()> Public Sub disConnect()
        Try
            SQL_SERVER.Close()
        Catch ex As Exception
            MsgBox(ex.Message.ToString, , "disConnect()")
        End Try
    End Sub

    <WebMethod()> Public Sub executeNonQuery(ByVal qryRec As String, ByVal status As
        Connect()
        'sqlCommand = New SqlCommand(FormatText(qryRec), SQL_SERVER)
        sqlCommand = New SqlCommand(qryRec, SQL_SERVER)
        sqlCommand.ExecuteNonQuery()
        If status.Trim <> "" Then
            MsgBox(status, MsgBoxStyle.Information, status)
        End If
        disConnect()
    Catch ex As Exception
        If InStr(ex.Message.ToString, "PRIMARY KEY") Then
            MsgBox("Given Value Already Found...", MsgBoxStyle.Exclamation, "Error")
        Else
            MsgBox(ex.Message.ToString)
        End If
    End Try
End Sub

<WebMethod()> Public Function getDataSet(ByVal qrySel As String) As
System.Data.DataSet
    Try
        Connect()
        dSet = New DataSet
        AdaSRC = New SqlDataAdapter(qrySel, sqlConStr)
        AdaSRC.Fill(dSet)
        disConnect()
    Catch ex As Exception
        MsgBox(ex.Message.ToString, , "getDataset()")
    End Try
    Return dSet

End Function

Public Function returnMax(ByVal qry As String) As Integer
    Dim max As Integer = 0
    Try
        Dim dset As New DataSet
        dset = getDataSet(qry)
        If dset.Tables(0).Rows.Count > 0 Then
            max = CInt(dset.Tables(0).Rows(0).Item(0).ToString.Trim)
        End If
    Catch ex As Exception

    End Try
    Return max
End Function

Public Function FormatText(ByVal strtext As String) As String
    Dim formatStr As String = ""
```

```
        Catch ex As Exception
            MsgBox(ex.Message.ToString, MsgBoxStyle.Critical, "FormatText()")
        End Try
        Return formatStr
    End Function
End Class
```

b) SQL

```
Imports System.Web
Imports System.Web.Services
Imports System.Web.Services.Protocols
Imports System.Data.OleDb
Imports System.Data

<WebService(Namespace:="http://tempuri.org/")> _
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Public Class WebService
     Inherits System.Web.Services.WebService
    Public sqlConStr As String = "provider=microsoft.jet.oledb.4.0; data
source=E:\Hospital1\hOSPITAL-Acces\Access\Hospital.mdb"
    'Public sqlConStr As String = "provider=microsoft.jet.oledb.4.0; data
source=\\192.168.1.5\F$\Ram\2010\ASP.Net\Hospital1\hOSPITAL-
Acces\Access\Hospital.mdb"
    'G:\Hospital1\hOSPITAL-Acces\Access
    Public SQL_SERVER As OleDbConnection = Nothing
    Dim sqlCommand As OleDbCommand
    Dim dSet As DataSet = Nothing
    Dim AdaSRC As OleDbDataAdapter
    Public displayQuery As String = ""

    <WebMethod()> Public Sub Connect()
        Try
            SQL_SERVER = New OleDbConnection(sqlConStr)
            If SQL_SERVER.State = ConnectionState.Open Then
                SQL_SERVER.Close()
                SQL_SERVER.Open()
            Else
                SQL_SERVER.Open()
            End If
        Catch ex As Exception
            MsgBox(ex.Message.ToString, , "Connect()")
        End Try
    End Sub
```

```
            SQL_SERVER.Close()
        Catch ex As Exception
            MsgBox(ex.Message.ToString, , "disConnect()")
        End Try
    End Sub

    <WebMethod()> Public Sub executeNonQuery(ByVal qryRec As String, ByVal status As
String)
        Try
            Connect()
            'sqlCommand = New SqlCommand(FormatText(qryRec), SQL_SERVER)
            sqlCommand = New OleDbCommand(qryRec, SQL_SERVER)
            sqlCommand.ExecuteNonQuery()
            If status.Trim <> "" Then
                MsgBox(status, MsgBoxStyle.Information, status)
            End If
            disConnect()
        Catch ex As Exception
            If InStr(ex.Message.ToString, "PRIMARY KEY") Then
                MsgBox("Given Value Already Found...", MsgBoxStyle.Exclamation, "Error")
            Else
                MsgBox(ex.Message.ToString)
            End If
        End Try
    End Sub

    <WebMethod()> Public Function getDataSet(ByVal qrySel As String) As
System.Data.DataSet
        Try
            Connect()
            dSet = New DataSet
            AdaSRC = New OleDbDataAdapter(qrySel, sqlConStr)
            AdaSRC.Fill(dSet)
            disConnect()
        Catch ex As Exception
            MsgBox(ex.Message.ToString, , "getDataset()")
        End Try
        Return dSet

    End Function

    Public Function returnMax(ByVal qry As String) As Integer
        Dim max As Integer = 0
        Try
            Dim dset As New DataSet
            dset = getDataSet(qry)
            If dset.Tables(0).Rows.Count > 0 Then
                max = CInt(dset.Tables(0).Rows(0).Item(0).ToString.Trim)
            End If
    End Try
    Return max
  End Function

  Public Function FormatText(ByVal strtext As String) As String
    Dim formatStr As String = ""
    Try
      formatStr = Replace(strtext, "'", "''")
    Catch ex As Exception
      MsgBox(ex.Message.ToString, MsgBoxStyle.Critical, "FormatText()")
    End Try
    Return formatStr
  End Function
End Class
```

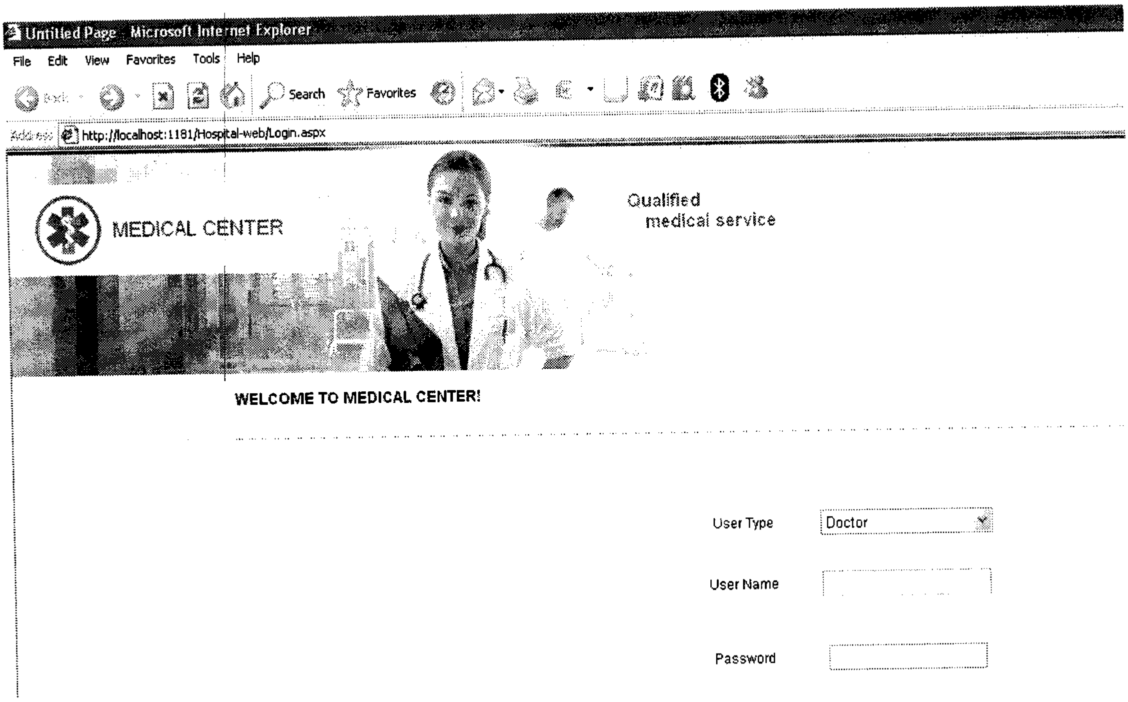
c)XML

```
Imports System.Web
Imports System.Web.Services
Imports System.Web.Services.Protocols
<WebService(Namespace:="http://tempuri.org/")> _
<WebServiceBinding(ConformsTo:=WsiProfiles.BasicProfile1_1)> _
<Global.Microsoft.VisualBasic.CompilerServices.DesignerGenerated()> _
Public Class XMLWeb
  Inherits System.Web.Services.WebService
  'G:\Hospital1\Hospital-XML\xml
  'Public sqlConStr As String = "\\192.168.1.5\F$\Ram\2010\ASP.Net\Hospital1\Hospital-
XML\xml\patient.xml"
  Public sqlConStr As String = "E:\Hospital1\Hospital-XML\xml\patient.xml"

  <WebMethod()> Public Function constr()

    Return sqlConStr

  End Function
End Class
```

APPENDIX II

SNAP SHOTS

LOG IN:



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
MEDICAL CENTER
Qualified
medical service

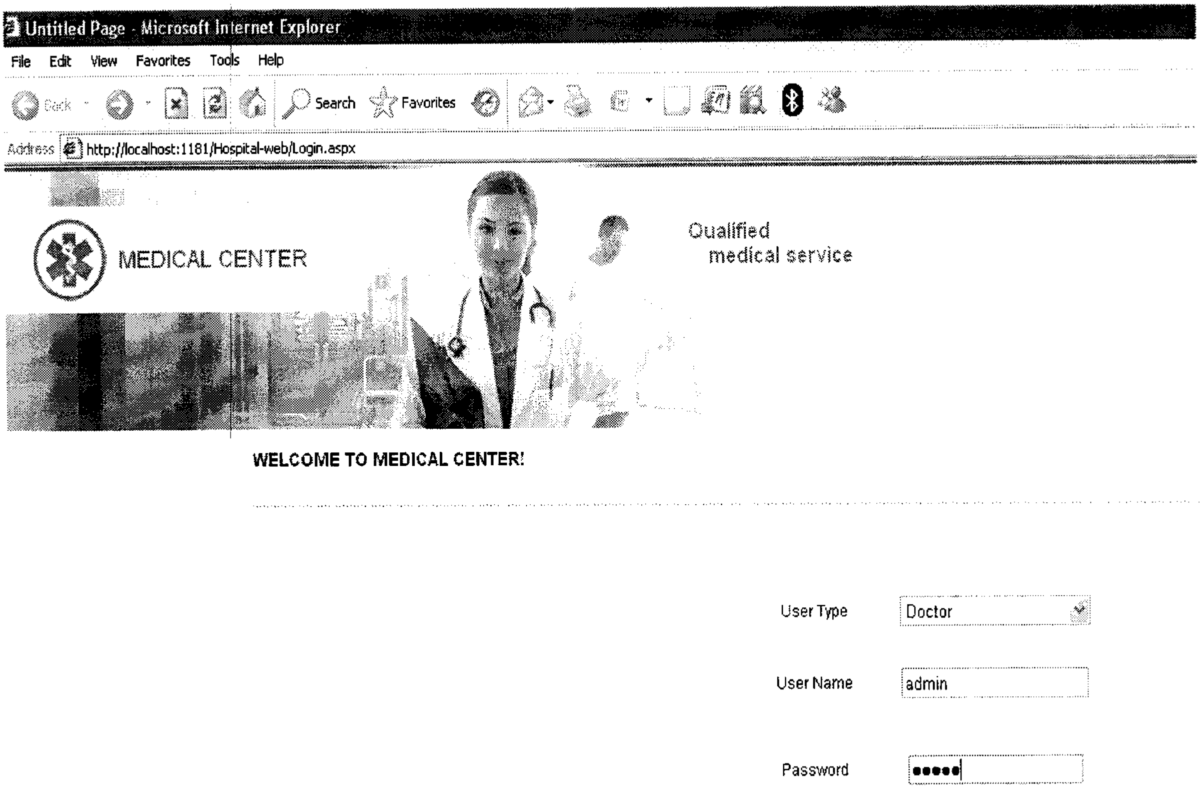
http://localhost:1181/Hospital-web/Login.aspx

WELCOME TO MEDICAL CENTER!

User Type: Doctor

User Name:

Password:


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
MEDICAL CENTER
Qualified
medical service

Address: http://localhost:1181/Hospital-web/Login.aspx

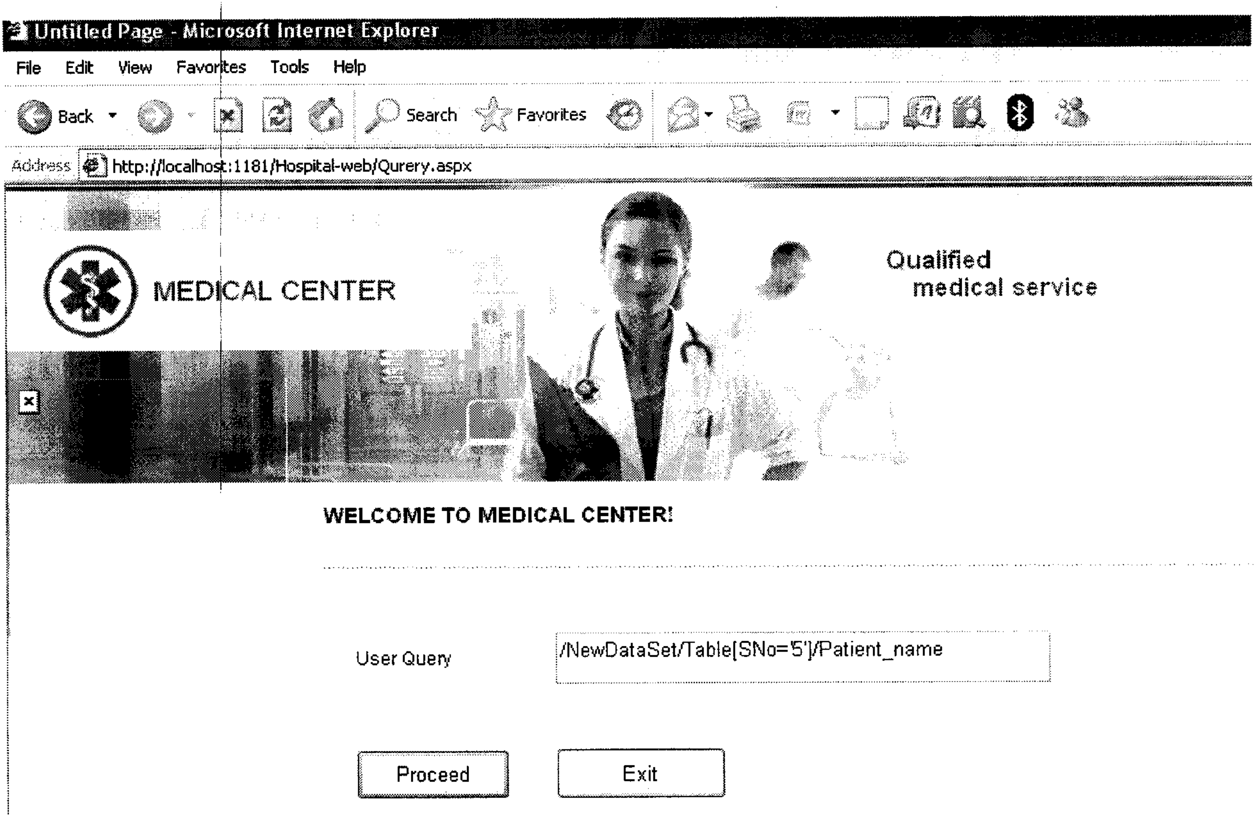
WELCOME TO MEDICAL CENTER!

User Type: Doctor

User Name: admin

Password: ●●●●●

USER QUERY:

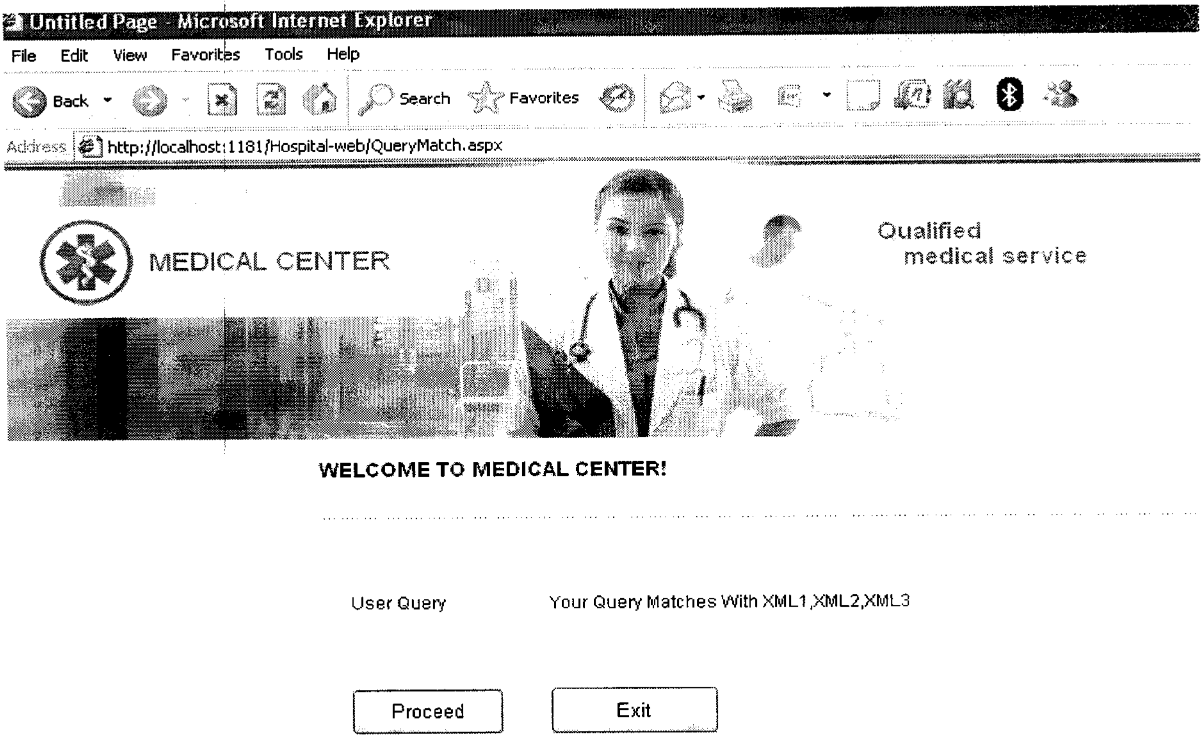

Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
MEDICAL CENTER
Qualified
medical service
Proceed
Exit

Address: http://localhost:1181/Hospital-web/Qurery.aspx

WELCOME TO MEDICAL CENTER!

User Query: /NewDataSet/Table[SNo='5']/Patient_name

SCHEMA MATCHING:

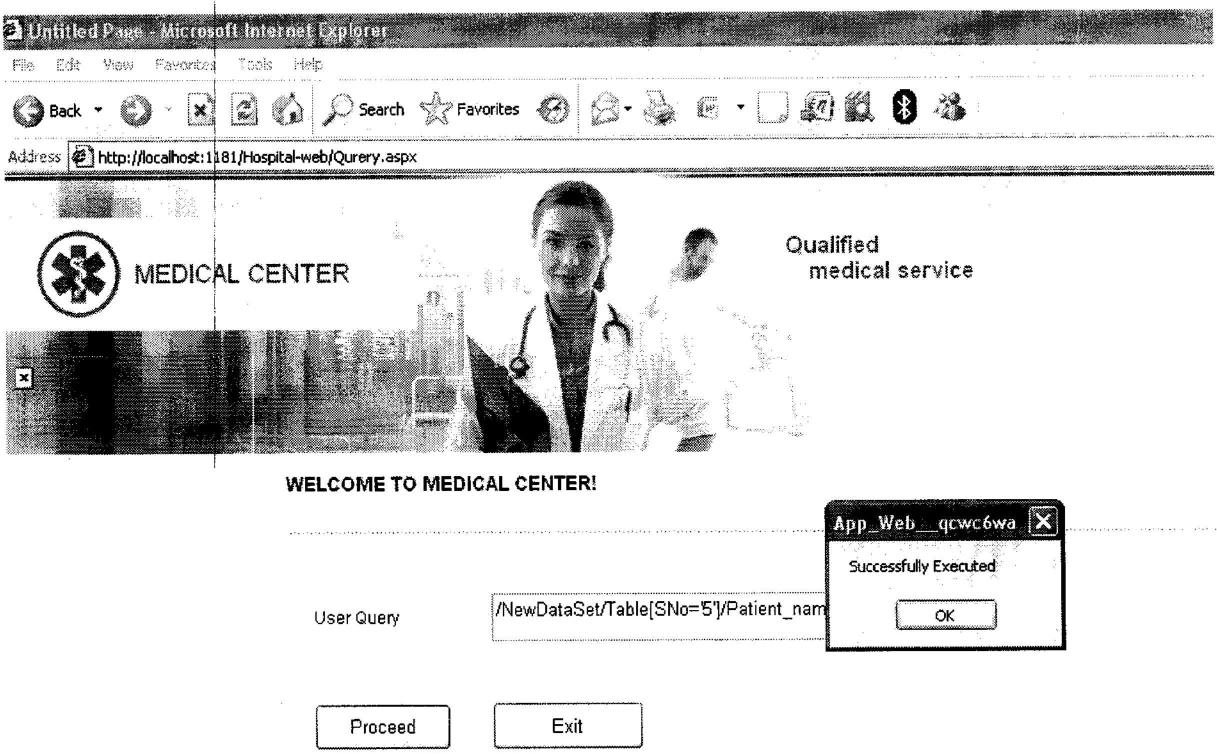

Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
MEDICAL CENTER
Qualified
medical service
Proceed
Exit

http://localhost:1181/Hospital-web/QueryMatch.aspx

WELCOME TO MEDICAL CENTER!

User Query Your Query Matches With XML1,XML2,XML3

RETRIEVING DATA FROM DIFFERENT SOURCE:



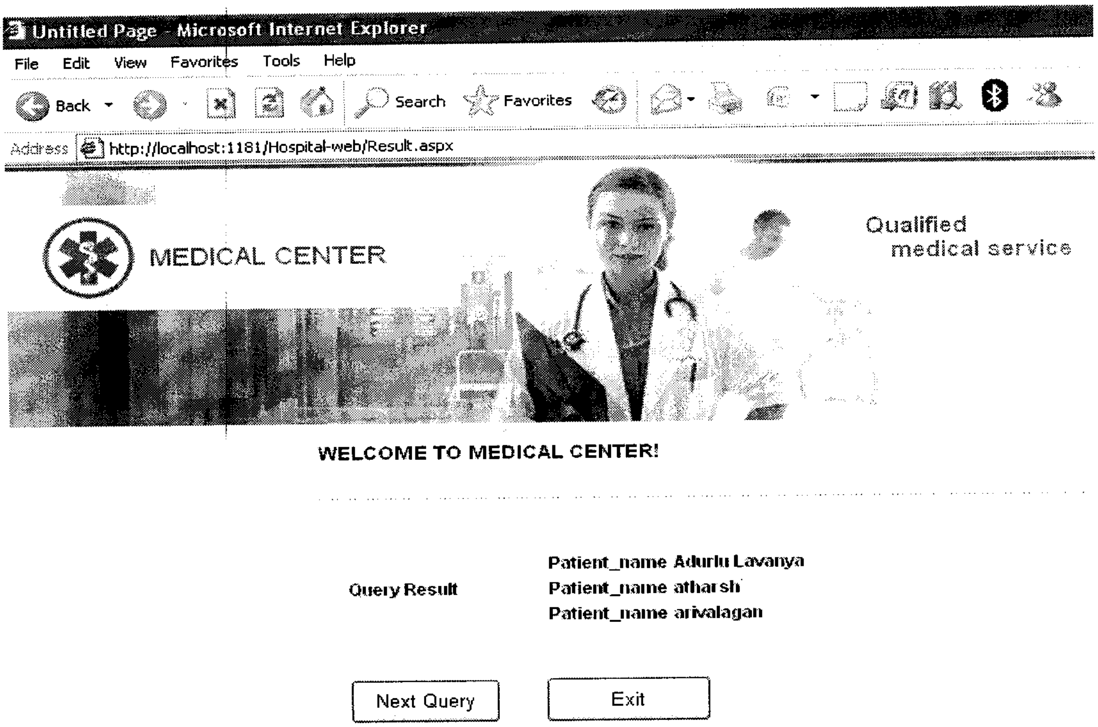
http://localhost:1181/Hospital-web/Qurery.aspx

WELCOME TO MEDICAL CENTER!

Successfully Executed

User Query /NewDataSet/Table[SNo='5']/Patient_nam

FINAL RESULT:


Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
MEDICAL CENTER
Qualified
medical service
Next Query
Exit

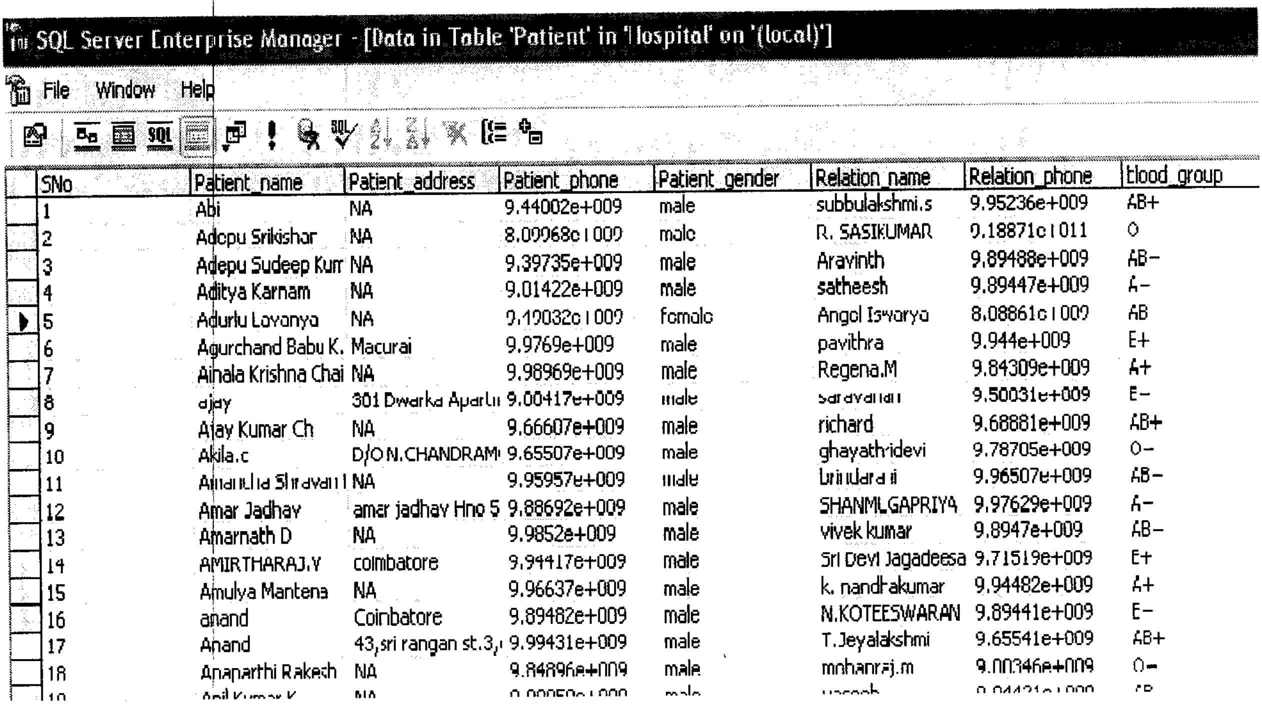
http://localhost:1181/Hospital-web/Result.aspx

WELCOME TO MEDICAL CENTER!

Query Result

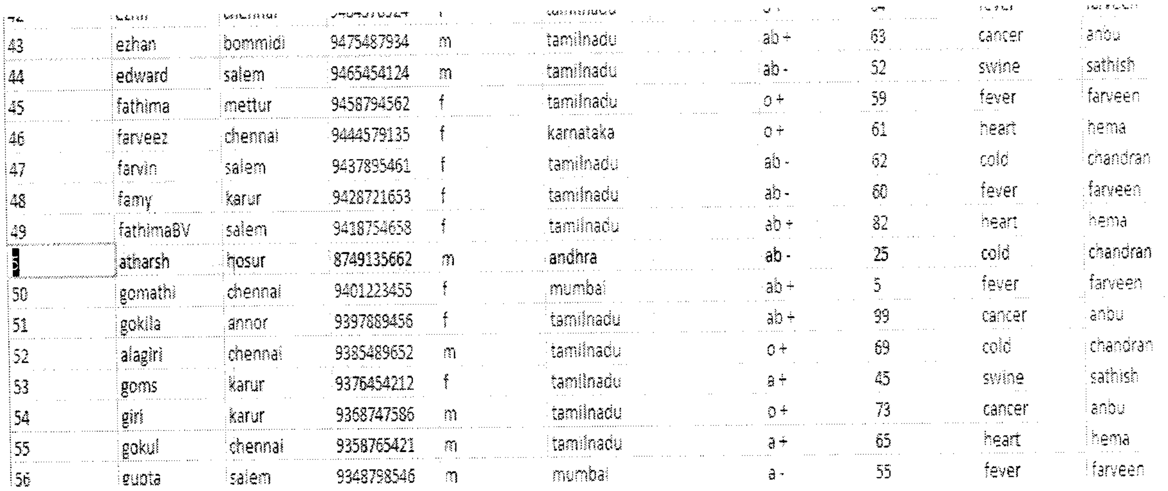
Patient_name Adurlu Lavanya
Patient_name atharsh
Patient_name arivalagan

SQL DATABASE:


SQL Server Enterprise Manager - [Data in Table 'Patient' in 'Hospital' on '(local)']
File Window Help

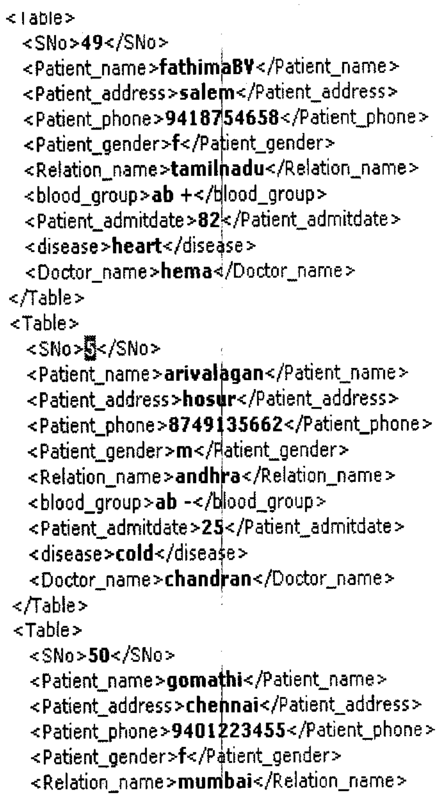
SNo	Patient_name	Patient_address	Patient_phone	Patient_gender	Relation_name	Relation_phone	Blood_group
1	Abi	NA	9.44002e+009	male	subbulakshmi.s	9.95236e+009	AB+
2	Adepu Srikishar	NA	8.09968e+009	male	R. SASIKUMAR	9.18871e+011	O
3	Adepu Sudeep Kum	NA	9.39735e+009	male	Aravinth	9.89488e+009	AB-
4	Aditya Karnam	NA	9.01422e+009	male	satheesh	9.89447e+009	A-
5	Adurlu Lavanya	NA	9.19032e+009	female	Angel Iswarya	8.08861e+009	AB
6	Agurchand Babu K.	Macurai	9.9769e+009	male	pavithra	9.944e+009	E+
7	Ainala Krishna Chai	NA	9.98969e+009	male	Regena.M	9.84309e+009	A+
8	ajay	301 Dwarka Apartn	9.00417e+009	male	saravanan	9.50031e+009	E-
9	Ajay Kumar Ch	NA	9.66607e+009	male	richard	9.68881e+009	AB+
10	Akila.c	D/O N.CHANDRAM	9.65507e+009	male	ghayathridevi	9.78705e+009	O-
11	Anirudha Shravani	NA	9.95957e+009	male	brindarani	9.96507e+009	AB-
12	Amar Jadhav	amar jadhav Hno 5	9.88692e+009	male	SHANMUGAPRIYA	9.97629e+009	A-
13	Amarnath D	NA	9.9852e+009	male	vivek kumar	9.8947e+009	AB-
14	AMIRTHARAJ.V	coimbatore	9.94417e+009	male	Sri Devi Jagadeesa	9.71519e+009	E+
15	Amulya Mantena	NA	9.96637e+009	male	k. nandhakumar	9.94482e+009	A+
16	anand	Coinbatore	9.89482e+009	male	N.KOTEESWARAN	9.89441e+009	E-
17	Anand	43,sri rangan st.3,	9.99431e+009	male	T.Jeyalakshmi	9.65541e+009	AB+
18	Ananarthi Rakesh	NA	9.84896e+009	male	mohanraj.m	9.00346e+009	O-
19	Anil Kumar K	NA	[illegible]	male	[illegible]	[illegible]	AB

ACCESS DATABASE:



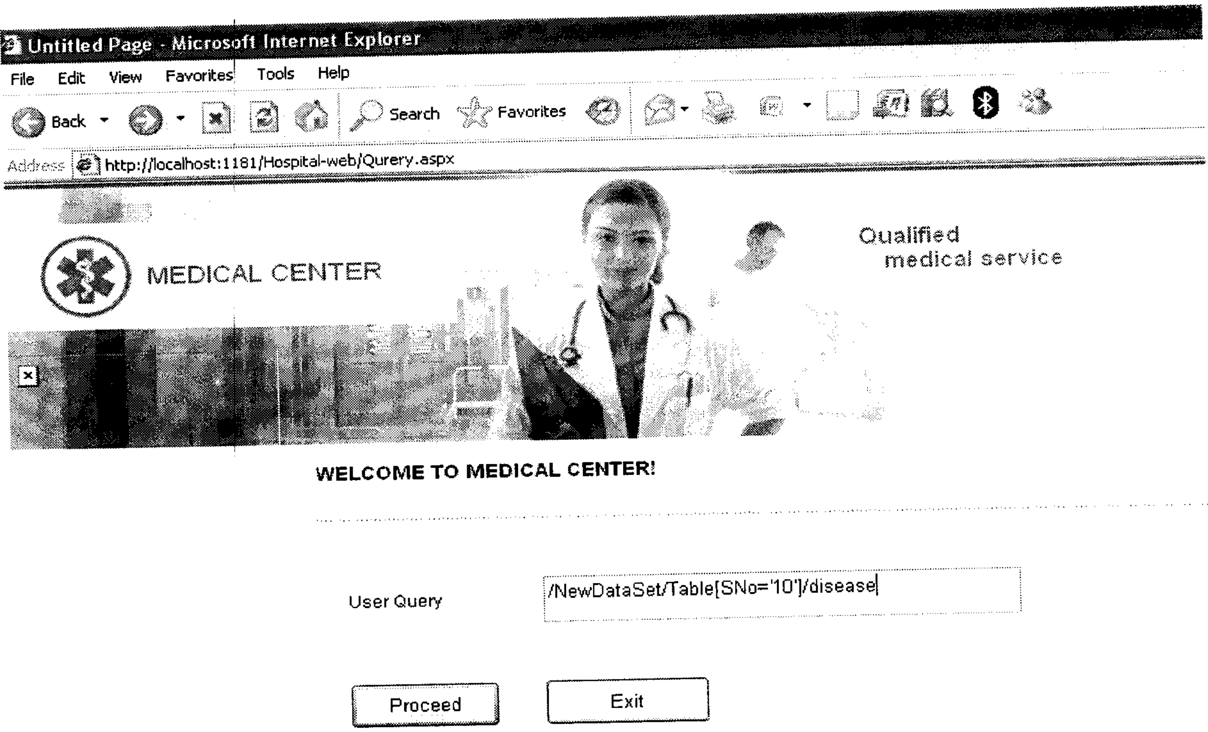
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
43	ezhan	bommidi	9475487934	m	tamilnadu	ab +	63	cancer	anbu
44	edward	salem	9465454124	m	tamilnadu	ab -	52	swine	sathish
45	fathima	mettur	9458794562	f	tamilnadu	o +	59	fever	farveen
46	farveez	chennai	9444579135	f	karnataka	o +	61	heart	hema
47	farvin	salem	9437895461	f	tamilnadu	ab -	62	cold	chandran
48	famy	karur	9428721653	f	tamilnadu	ab -	60	fever	farveen
49	fathimaBV	salem	9418754658	f	tamilnadu	ab +	82	heart	hema
5	atharsh	hosur	8749135662	m	andhra	ab -	25	cold	chandran
50	gomathi	chennai	9401223455	f	mumbai	ab +	5	fever	farveen
51	gokila	annor	9397889456	f	tamilnadu	ab +	99	cancer	anbu
52	alagiri	chennai	9385489652	m	tamilnadu	o +	69	cold	chandran
53	goms	karur	9376454212	f	tamilnadu	a +	45	swine	sathish
54	giri	karur	9368747586	m	tamilnadu	o +	73	cancer	anbu
55	gokul	chennai	9358765421	m	tamilnadu	a +	65	heart	hema
56	gupta	salem	9348798546	m	mumbai	a -	55	fever	farveen

XML DATABASE:



```
<Table>
  <SNo>49</SNo>
  <Patient_name>fathimaBV</Patient_name>
  <Patient_address>salem</Patient_address>
  <Patient_phone>9418754658</Patient_phone>
  <Patient_gender>f</Patient_gender>
  <Relation_name>tamilnadu</Relation_name>
  <blood_group>ab +</blood_group>
  <Patient_admitdate>82</Patient_admitdate>
  <disease>heart</disease>
  <Doctor_name>hema</Doctor_name>
</Table>
<Table>
  <SNo>5</SNo>
  <Patient_name>arivalagan</Patient_name>
  <Patient_address>hosur</Patient_address>
  <Patient_phone>8749135662</Patient_phone>
  <Patient_gender>m</Patient_gender>
  <Relation_name>andhra</Relation_name>
  <blood_group>ab -</blood_group>
  <Patient_admitdate>25</Patient_admitdate>
  <disease>cold</disease>
  <Doctor_name>chandran</Doctor_name>
</Table>
<Table>
  <SNo>50</SNo>
  <Patient_name>gomathi</Patient_name>
  <Patient_address>chennai</Patient_address>
  <Patient_phone>9401223455</Patient_phone>
  <Patient_gender>f</Patient_gender>
  <Relation_name>mumbai</Relation_name>
```

USER QUERY:



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:1181/Hospital-web/Qurery.aspx

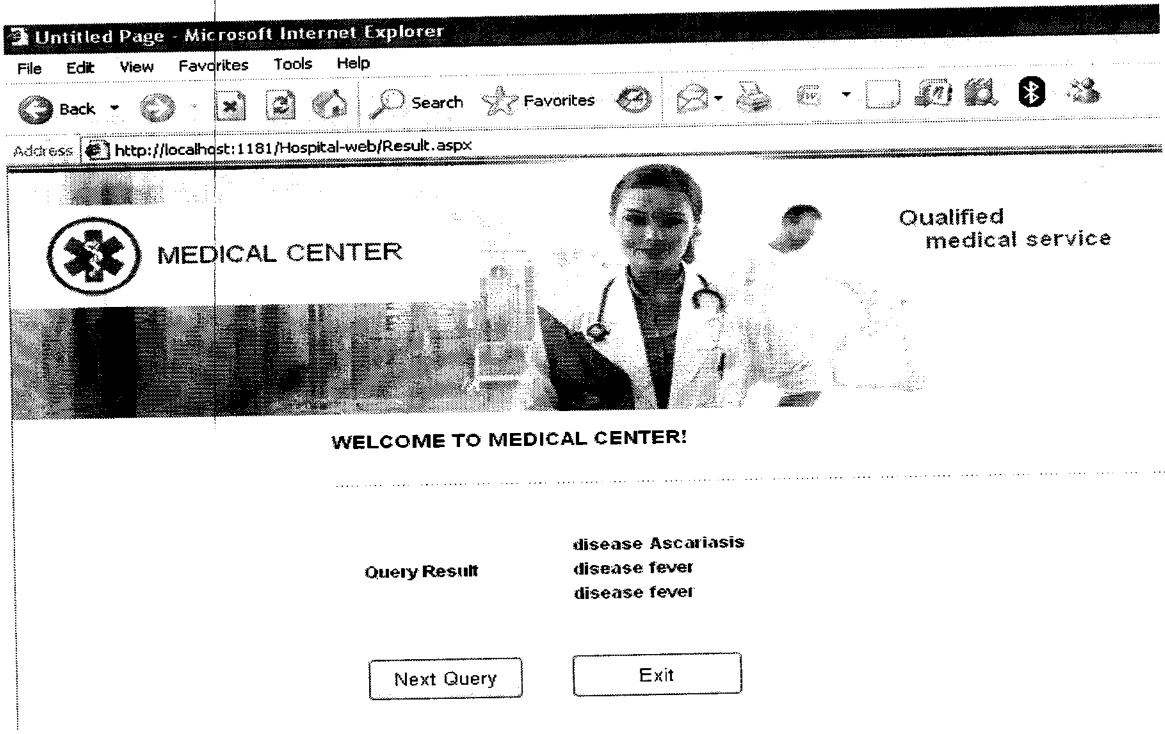
MEDICAL CENTER

Qualified medical service

WELCOME TO MEDICAL CENTER!

User Query /NewDataSet/Table[SNo='10']/disease

FINAL RESULT:



Untitled Page - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:1181/Hospital-web/Result.aspx

MEDICAL CENTER

Qualified medical service

WELCOME TO MEDICAL CENTER!

Query Result disease Ascariasis
disease fever
disease fever

REFERENCES

1. AMotro, (1987) 'Virtual Integration of Multiple Databases', IEEE IS. Vol13, 7.

2. APSheth, J.ALarson, 'Federated Database Systems for managing distributed heterogeneous and autonomous databases.', ACM Computing Surveys.

3. Alonso. R., Barbara D. (1989). 'Negotiating data integration using federated database systems', In Proceedings of the 5th International Conference on Data Engineering (Feb.), pp. 56-65.

4. Bressan, S., Goh, C., Levina, S., Madnick, S., Shah, A., and Siegel, M.,(2002) 'Context Knowledge Representation and Reasoning in the Context Interchange System', Applied Intelligence (13:2), pp. 165-179.

5. Bohm K, Aberer K.T, Ozsu M, Gayer K,(2008) 'Query Optimization for structured documents based on knowledge on the document type definiton', IEEE International forum on Research and Technology Advances in Digital Libraries.

6. B. Medjahed, A. Bouguettaya.(2003) 'Composing web services on the semantic web', Journal on Very Large Databases, vol. 4 , pp. 333-351.

7. Carey,M.Florescu (2003) 'Integrating Enterprise Information on on-demand with XQuery',XML Journal,Vol,3,NOs 6 and 7.

8. David, P. A. (1995) 'Standardization Policies for Network Technologies: The Flux Between Freedom and Order Revisited'. In Hawkins, R., Mansell, R., and Skea, J. (eds): Standards, Innovation and Competitiveness,Edward Elgar Publishers, pp. 15-35.

9. D.D. Duong and V. Wuwongse,(2003) 'XML Database Schema Integration Using XDD', Proceedings of Advances in Web-Age Information Management Conference, China: Lecture Notes in Computer Science, Springer Verlag, vol. 2762, pp. 92-103.

10. Franceschet, M., XPathMark – (2005) 'An XPath benchmark for XMark'. Research report PP-, University of Amsterdam, The Netherlands.

11. Fielding, R., Taylor, R.N. (2000) 'Principled design of the modern Web architecture'. ACM Press New York, 407-416.

12. G. J. Bex, S. Maneth, and F. Neven. (2000) 'A Formal Model for an Expressive Fragment of XSLT'. In Proc. CL 2000, LNCS 1861, pages 1137–1151. Springer.

13. HEIMBIGNER, D., AND MCLEOD, D. (1985) 'A federated architecture for information management'. ACM Trans. Off. Znf. Syst. 3, 253-278.

14. J. Dzikiewicz and C. Egyhazy. Cyrano (1994)' A Meta Model for Federated Database Systems' In Proceedings of ISMM International Conference, pp. 52-55.
15. L. Zeng, B. Benatallah and M. Dumas, 'Data Consolidation," Proceeding of the 12th International Conference on World Wild Web', pp. 441-421, ACM.

16. Lakshmanan L., F. Sadri, S. Subramanian: (1997)'Logic and Algebraic Languages for Interoperability in Multidatabase Systems'. JLP 33(2): 101-149

17. Mello, R.S.; Heuser, C.A. (2001) 'A Bottom-Up Approach for Integration of XML Sources.In': International Workshop on Information Integration on the Web (WIIW'2001),pp. 118-124, Rio de Janeiro, Brazil.

18. Michael Benedikt, Wenfei Fan, Gabriel M. Kuper: (2003), 'Structural Properties of XPath Fragments'. 79-95.

19. Paul Patrick, Impact of SOA on Enterprise Information Architectures, SIGMOD ACM, 2005,June, 844-848

20. P. M. D. Gray, S. M. Embury, K. Y. Hui, and G. J. L. Kemp (1999), 'The EII Layer',J. Intell. Inform.Syst, vol. 12, pp. 113–137.

21. S. Aissi, P. Malu and K. Srinivasan, (2002) 'Data Integration: the Next Big Step,' Computer, IEEE, vol. 35, iss. 5,DOI:10.1109/MC.., pp.55–62.

22. Yinzhang GUO,Gangzhu QIAO,Yubin XU,(2004) 'An Reused Enterprise Information System Architecture Based on CCM Business Component' Journal of computer science (Chinese),31(9)

23. Yueguo Chen,Jingchun Wang. (2004) 'The Summary of Data integration'. Computer Science, 38 (5): 48-51.

24. Z. Luo and J.S. Li, (2005) 'Data Propogation,' Proceeding of the 2005 International Conference on Services Computing, Vol. 2, pp: 245-248, IEEE.

25. Zhaohui Wu, Huajun Chen, Yuxing Mao, Guozhou Zheng,(2005) 'Dart Database Grid: A Dynamic, Adaptive, RDF- mediated,Transparent Approach to Database Integration for Semantic Web', APWeb: 1053-1057.