University of Pretoria etd – Rimer, S C (2006)

Chapter 1 : RESEARCH OVERVIEW

1.1 Introduction

This chapter describes the current problems businesses are increasingly facing when using inter-networked computer systems to support multiple applications. It defines the problem statement, the research objectives, the research approach and the scope of the work undertaken.

1.2 Scope

The scope of the research is to provide a generic XML interface to multiple client server applications using Internet type transport protocols such as HTTP and TCP. It does not focus on any security concerns especially with respect to the transportation of data over an open network. It assumes that there will be some sort of access control mechanism in place on a client server application and therefore that some authentication information will have to be provided to the server applications.

1.3 Problem Statement

Protocols such as LDAP or applications that use a relational database operate using a client-server type model where the client requests information from a server, and the server returns a response to the client.

Applications using different protocols typically require support for each of the protocols to be implemented on a separate front-end component. Application interaction (via messages) is achieved by using separate custom built front-end applications with non-portable data formats and functionality.

Traditionally, a client connected to the Internet that requires access to different server applications would have protocol specific programs at the client to access the data. When a new server application is added to the system, a separate client application is developed to access the new protocol's data on the host server.

University of Pretoria etd – Rimer, S C (2006)

For example consider a typical system connected to the Internet that has the following applications/services running on it:

- A directory service (i.e. LDAP).
- A TCP based proprietary server program.
- A relational database server.

The traditional architecture would require separate client programs accessing each of the server or database applications as shown in the figure below.

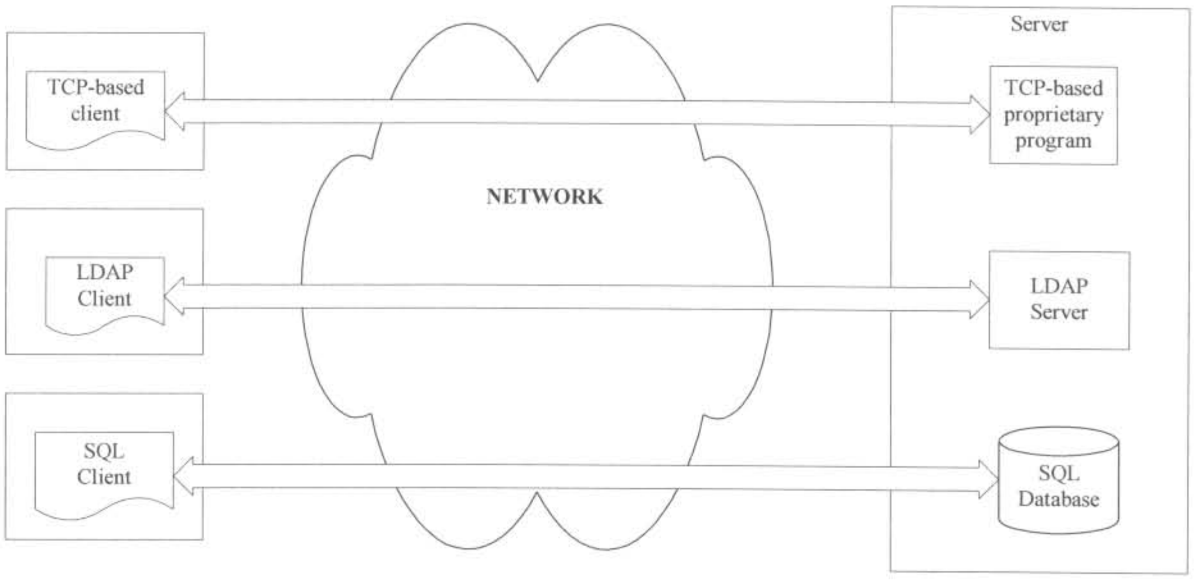

Figure 1: Traditional architecture requiring separate client applications

The system architecture depicted in Figure 1 increases the complexity on the client's side, as it needs to have the latest version of each specific protocol program installed on its workstation. In addition, if a new server application is added, each client needs to load another client program that can access the server's data. The need to install specialised clients on workstations increases the complexity of maintaining a system and reduces flexibility to introduce new protocols into a system. Any addition of new software may require the addition of new-shared libraries that may complicate or interfere with existing applications (such as stability, versions etcetera).

University of Pretoria etd – Rimer, S C (2006)

A need exists to access information from different heterogeneous systems in a standard message request-response format. The rules surrounding request-response type messages and their data should be organized in a clear and consistent manner, so that information can be shared among many applications.

The eXtensible Markup Language (XML) is proposed as a solution to access diverse information systems through a common metadata model. XML is particularly suited to web-based data exchange because it allows data exchange across disparate platforms and operating systems.

This project describes an XML model that abstracts the differences in underlying heterogeneous client-server systems and provides a common XML message interface.

To design the model the similarities of different client-server applications were identified, and using these similarities a common messaging system using XML as the command interpretation language was developed.

Figure 2 provides a high-level overview of the proposed system architecture.

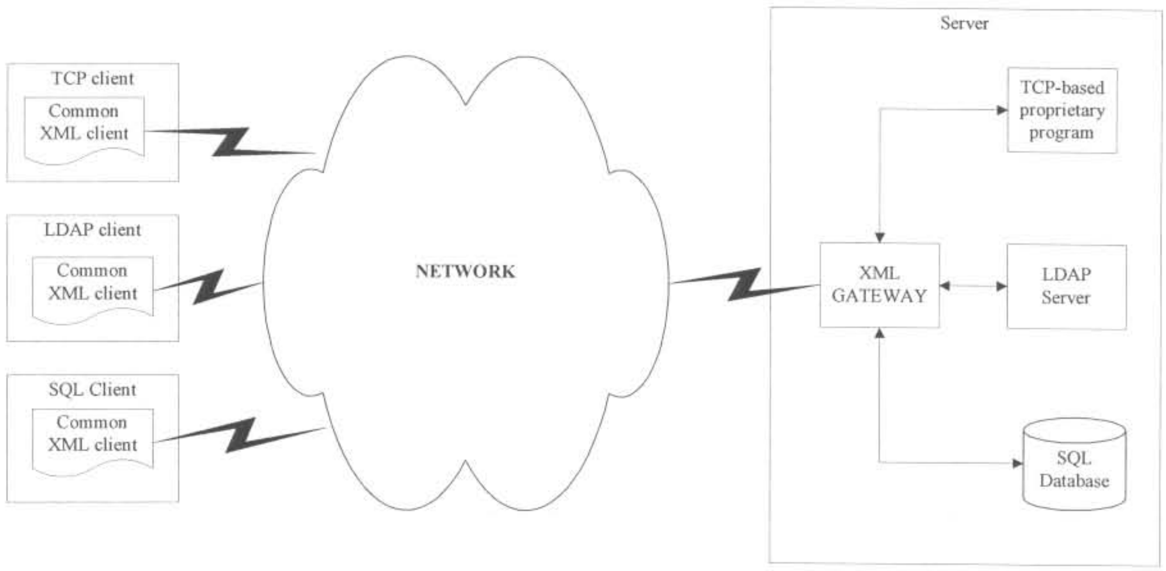

Figure 2: Proposed architecture depicting protocol-to-XML clients interfacing to an XML gateway

University of Pretoria etd – Rimer, S C (2006)

1.4 Research Context

Studies on the problem of restructuring and reformatting of data as it passes from one software tool or process to another predates the widespread use of the Internet that started in the mid 1990s. Blattner et al. [18,19] in a study on generic message translation, attempted to solve the problem by providing a visual interface that can create a mapping between fields in different message types that specifies which fields have similar semantic content. The Blattner et al. papers were published before the introduction of XML into the computing landscape. However, in their paper, the authors conclude that some sort of "parser-generator" must be constructed to take descriptions for data specifications and create a "translator" between systems.

Since the introduction of XML, several studies have been undertaken on the feasibility of using XML, such as the article by Bi et al. [15] that focuses on using XML to interact with multiple legacy applications, and an article by Peinl and Mitchang [20], which investigates transforming independent, autonomous data sources into a common XML format in order to provide an integrated communication platform for mobile applications. Both articles acknowledge the advantages provided by using XML as the data modelling and exchange mechanism between applications (clients) and information sources (servers).

However, use of XML for generic messaging does not come without some disadvantages, namely slower processing speed caused by the additional overhead of using XML to transform and parse messages. Boedjang et al. [21] in their study of distributed data structures conclude that the performance measurements of applications that run application-specific code are faster than those that use generic message passing software.

1.5 Research Objective

The main objective of this research is to abstract the differences in underlying heterogeneous systems by designing a generic XML component that provides a common client that can interface with multiple heterogeneous client-server applications.

To achieve this, a number of sub-objectives were examined. These sub-objectives are:

- Examine the current process in which the individual client-server applications (such as

University of Pretoria etd – Rimer, S C (2006)

LDAP, SQL and ESD) interact,

- Investigate current implementations of XML-LDAP and XML-SQL,
- Determine XML strengths and weaknesses,
- Identify advantages, if any in using XML for a common messaging system for multiple heterogeneous applications,
- Investigate methods of error handling within the XML gateway component,
- Investigate using SOAP as the method of communicating with both client and server devices and with other management entities,
- Identify problems in providing a single messaging interface to multiple applications,
- Analyse any existing generic XML interfaces that may be similar to area of research undertaken,
- Investigate the current functions and objects used in the existing applications that will have to be implemented in the XML gateway,
- Develop a generic XML schema that correctly models the commands and data of the different applications.
- Design and develop a generic XML front-end that allows current Internet protocols to access the gateway services through the same application, and
- Design and develop a generic XML gateway for use by multiple client-server applications.

1.6 Research Approach

This section identifies the main questions that need to be answered and describes the research approach used in attempting to answer these questions.

1.6.1 Research Questions

The following problem solving questions were identified to assist in better understanding and defining the problem.

1. What is the current means of communication in terms of command structure and message format between each of the multiple client-server applications?
2. What commonalities do these diverse applications share in terms of command structure

University of Pretoria etd – Rimer, S C (2006)

and data format?

3. What advantages will be derived from using XML as the common interface language?
4. What disadvantages will arise from using a single XML interface to multiple applications?
5. What are the application boundaries?
6. Why use XML?
7. How will the various technologies and applications interact?
8. What are the system resource constraints (if any) on the application?
9. How will the XML gateway application decide which of the different applications to send a request to?
10. Does XML provide sufficient benefit to overcome its shortcomings?

1.6.2 Research Instruments

1. Literature Study:
    a. The scope of the work was identified.
    b. A literature study was undertaken to understand each applications message and data structure used in client-server interaction.
    c. A literature study was undertaken to investigate existing XML standards (if any) for LDAP and SQL. Note ESD is a proprietary protocol and there is no previous XML framework/standard for it.
    d. A literature study was undertaken to understand the structure and syntax of XML.
    e. A literature study was undertaken to understand the structure and syntax of the XML DTD and the XML Schema.
    f. A search was done to identify appropriate open source applications that would serve as an LDAP server, a relational database, a web server and an XML parser.

2. Problem Solving Analysis
    a. An analysis of each of the applications message and data structure was undertaken and certain commonalities in commands and parameters were

University of Pretoria etd – Rimer, S C (2006)

identified.

b. An analysis of existing LDAP-XML and SQL-XML frameworks was conducted to determine if they were suitable for the needs of the study/research problem.

c. An analysis of XML DTD's and XML schema was undertaken to determine which would be the most appropriate to define the designed common data model.

3. Design
    a. An XML schema was developed that models the request-response command structures of client server applications in a generic way that can be used across multiple client-server type applications.
    b. An addressing mechanism was devised to identify which application the request is destined for.
    c. An XML gateway was designed to send requests to the correct destination application using a structure that allows for easy addition of new applications.
    d. A front-end interface was designed to provide a GUI to users to send requests to multiple applications and to view the corresponding responses in a user-friendly interface.
    e. Several test programs were written to test individual components of the design.

4. Implementation
    a. The open source applications that could serve as an LDAP server, a database, a web server and an XML parser were set up to run on the specified operating system.
    b. The design was implemented using an appropriate programming language.
    c. The design was tested using several test programs.
    d. The complete design was tested using the implemented web front-end component.

5. Analysis and Assessment
    a. A final analysis of the system in terms of, meeting functional requirements, and

University of Pretoria etd – Rimer, S C (2006)

performance was undertaken.

b. The results were provided and an analysis of the results in terms of the above-mentioned research objectives and questions is provided.

c. Final conclusions of the results are provided.