

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

**DESIGN OF A GENERIC
CLIENT-SERVER MESSAGING INTERFACE
USING XML**

by

Suvendi Chinnappen Rimer

Submitted in partial fulfilment of the requirements for the degree
Master of Engineering (Computer Engineering)
in the
Faculty of Engineering
UNIVERSITY OF PRETORIA

March 2003

DISSERTATION SUMMARY

**DESIGN OF A GENERIC
CLIENT-SERVER MESSAGING INTERFACE
USING XML**

by

Suvendi Chinnappen Rimer

Study Leader: Prof. G. P. Hancke
Department: Electrical, Electronic and Computer Engineering
UNIVERSITY OF PRETORIA
Degree: MEng (Computer Engineering)

Applications that use directory services or relational databases operate in client-server mode where a client requests information from a server, and the server returns a response to the client. Communication between each client-server application is achieved by using separate custom built front-ends with non-portable data formats. A need exists to access information from different heterogeneous client-server systems in a standard message request-response format.

This research proposes a generic XML document that presents a common request-response interface to the client from which they can access network protocol or database information. The XML component is easily adaptable to accessing any new client-server type protocol or database data that may be added to a server.

The approach in determining the XML elements is, firstly review each systems command and data structure separately, and then determine if there are any commonalities within each protocol that would allow for a common representation of both the data and command structure.

For the purposes of this project, three different data sources that are typically used in an Internet application were analysed, namely:

- A TCP based server program.
- A relational type database.
- A directory service.

The solution was implemented using Linux as the operating system, Java as the programming language, MySQL as the relational database, openLDAP as the directory server and a proprietary TCP based server application. Initially the complete system was developed for the proprietary TCP-based application. The other systems were added with minimum additional work.

The result of the implementation was that it is relatively easy to add new protocols (for e.g. LDAP) on an as needed basis with minimal changes required on the server side. A client will receive XML responses that the client can either adapt (typically using a separate style-sheet) to their specific needs or use the existing front-ends if they are suitable.

After the design was implemented and tested, the performance of XML and non-XML messages was evaluated. As expected the increased verbosity of XML results in a larger footprint that requires more processing time and resources. This means that any implementation using XML has to carefully weigh the benefits of flexibility, extensibility and standard message formats against reduced performance.

After evaluating XML type messages in an Internet type environment that involved human-computer interaction, it was concluded that the slower response times is not that significant to negate the benefits of a common message interface provided by using XML.

SAMEVATTING VAN VERHANDELING

**DIE ONTWERP VAN ‘N GENERIESE
BOODSKAP-INTERVLAK VIR ‘N KLIËNTBEDIENER-STELSEL MET BEHULP
VAN XML
deur
Suvendi Chinnappen Rimer**

Studieleier: Prof G.P. Hancke
Departement: Elektriese, Elektroniese en Rekenaar-Ingenieurswese
UNIVERSITEIT VAN PRETORIA
Graad: MIng (Rekenaar-Ingenieurswese)

Toepassings wat gidisdienste of verhoudingsdatabasisse gebruik, werk in ‘n kliëntbedienermodus waar die klient inligting van die bediener versoek en die bediener ‘n antwoord na die klient terugstuur. Kommunikasie tussen elke kliëntbediener-toepassing vind plaas deur die gebruik van afsonderlike toegewyde intervlakke met nie-oordraagbare data-formate. ‘n Behoefte bestaan om toegang te verkry tot inligting vanaf verskillende heterogene kliëntbediener-stelsels deur ‘n standaard versoek-antwoord-formaat te gebruik.

Hierdie navorsing stel ‘n generiese XML-dokument voor wat ‘n algemene versoek-antwoord-intervlak vir die kliënt aanbied waarvandaan kliënte toegang het tot netwerk-protokol- of databasis-inligting. Die XML-komponent is maklik aanpasbaar vir nuwe Internet-tipe protokolle of databasis-inligting wat tot die bediener bygevoeg word.

Die benadering tot die bepaling van die XML-elemente was om elke stelselopdrag en datastruktuur afsonderlik te beskou en dan te bepaal watter ooreenkomste binne die protokolle bestaan wat ‘n gemeenskaplike voorstelling van beide die data- en die bevelstruktuur moontlik maak.

Vir die doel van hierdie navorsing is drie verskillende bronne van data wat tipies in Internet-toepassings gebruik word, ontleed:

- 'n TCP-gebaseerde bediener-program
- 'n verhoudingsdatabasis
- 'n gidssdiens

Die oplossing is geïmplementeer deur Linux as die bedryfstelsel, Java as die programmeringstaal, MySQL as die verhoudingsdatabasis en openLDAP as die gidssdiens te gebruik. Eerste is die totale stelsel vir die TCP-gebaseerde toepassing ontwikkel en die ander twee toepassings is sonder veel bykomende werk bygevoeg.

Die bevinding tydens die implementering was dat dit maklik is om nuwe protokolle (bv. LDAP) by te voeg met minimale veranderings aan die bediener se kant. 'n Kliënt sal XML-antwoorde kry wat die kliënt kan aanpas vir sy spesifieke behoeftes of slegs deur die beskikbare intervlakke te gebruik.

Nadat die ontwerp geïmplementeer en getoets is, is die gedrag van XML teen nie-XML boodskappe geëvalueer. Soos verwag is, het die beter woordrykheid van die XML 'n groter voetspoor tot gevolg wat meer prosesseertyd en hulpbronne benodig. Dit beteken dat enige implementering wat XML gebruik, die voordele van aanpasbaarheid, uitbreikbaarheid en standaard boodskapformate teen verlaagde werkverrigting moet afspeel.

Nadat die gebruik van XML-tipe boodskappe beoordeel is in 'n Internet-omgewing waar mens-rekenaar interaksie betrokke is, is tot die gevolgtrekking gekom dat die voordele van die gemeenskaplike boodskapintervlak, deur die gebruik van XML, groter is as nadeel van die effens stadiger reaksieertyd.

List of Abbreviations

CIM	Common Information Model
DAP	Directory Access Protocol
DIT	Directory Information Tree
DN	Distinguished Name
DTD	Document Type Definition
DSML	Directory Services Markup Language
ESD	Extended Services Daemon
HTTP	HyperText Transport Protocol
LDAP	Lightweight Directory Access Protocol
OSI	Open Systems Interconnection
RDN	Relative Distinguished Name
SGML	Standard Generalized Markup Language
SOAP	Simple Object Access Protocol
SQL	Structure Query Language
TCP	Transmission Control Protocol
XML	eXtensible Markup Language
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery and Integration

Table of contents

<u>Chapter 1</u>	<u>: RESEARCH OVERVIEW</u>	1
1.1	<u>Introduction</u>	1
1.2	<u>Scope</u>	1
1.3	<u>Problem Statement</u>	1
1.4	<u>Research Context</u>	4
1.5	<u>Research Objective</u>	4
1.6	<u>Research Approach</u>	5
1.6.1	<u>Research Questions</u>	5
1.6.2	<u>Research Instruments</u>	6
<u>Chapter 2</u>	<u>: LITERATURE STUDY</u>	9
2.1	<u>Overview of application technology</u>	9
2.1.1	<u>Lightweight Directory Access Protocol (LDAP)</u>	9
2.1.2	<u>Databases and SQL</u>	14
2.1.3	<u>Differences Between Directories and Databases</u>	17
2.1.4	<u>The IGUANA gateway</u>	17
2.1.5	<u>Extended Service Daemon (ESD)</u>	18
2.1.6	<u>XML</u>	25
2.2	<u>Related Work</u>	33
2.2.1	<u>LDAP and XML: Directory Services Markup Language (DSML)</u>	33
2.2.2	<u>SQL and XML: SQLX (INCITS)</u>	33
2.2.3	<u>Common Information Model (CIM)</u>	34
2.2.4	<u>Simple Object Access Protocol (SOAP)</u>	35
2.3	<u>Related Work in published Articles.</u>	38
<u>Chapter 3</u>	<u>: SYSTEM OVERVIEW</u>	40
3.1	<u>Existing System</u>	40
3.2	<u>Proposed System</u>	42
<u>Chapter 4</u>	<u>: SYSTEM SPECIFICATIONS</u>	44
4.1	<u>Introduction</u>	44
4.2	<u>System Requirements</u>	44
4.3	<u>System Constraints</u>	45
4.4	<u>Assumptions</u>	45
4.5	<u>Data Structure</u>	45
4.6	<u>Functional Specification</u>	46
4.7	<u>Software Methodology</u>	48
4.8	<u>Testing</u>	49
4.9	<u>External Software Components</u>	49
4.9.1	<u>Database</u>	50
4.9.2	<u>Directory Server</u>	50
4.9.3	<u>Web Server</u>	50
4.9.4	<u>XML Parser</u>	51
4.9.5	<u>XML Translator</u>	51
4.9.6	<u>Programming Language</u>	52
4.9.7	<u>Operating system</u>	52
<u>Chapter 5</u>	<u>: XML MODEL</u>	53
5.1	<u>Identification of similarities between the heterogeneous applications</u>	53
5.2	<u>XML Document Design</u>	53

<u>Chapter 6</u>	<u>: APPLICATION ARCHITECTURE</u>	59
6.1	<u>Application Architecture</u>	59
6.1.1	<u>Presentation Layer</u>	61
6.1.2	<u>Business Layer</u>	62
6.1.3	<u>Data Layer</u>	63
6.2	<u>Interfaces to External Applications</u>	63
<u>Chapter 7</u>	<u>: SOFTWARE DESIGN</u>	65
7.1	<u>Introduction</u>	65
7.1.1	<u>The package diagram</u>	65
7.1.2	<u>The class diagram</u>	65
7.1.3	<u>The flowchart</u>	65
7.2	<u>Design</u>	66
7.2.1	<u>The Presentation layer</u>	66
7.2.2	<u>The business layer</u>	70
7.2.3	<u>The Data layer</u>	82
7.3	<u>General library classes</u>	85
7.4	<u>Application flowchart</u>	86
7.5	<u>Error Handling</u>	88
<u>Chapter 8</u>	<u>: RESULTS AND ANALYSIS</u>	89
8.1	<u>Results</u>	89
8.1.1	<u>ESD Results</u>	89
8.1.2	<u>SQL Results</u>	90
8.1.3	<u>LDAP Results</u>	91
8.1.4	<u>Results of different application servers</u>	91
8.1.5	<u>Measuring the relationship between number of XML messages and time.</u>	92
8.2	<u>Analysis of Results</u>	96
8.2.1	<u>Comparison of XML vs. non-XML type messages</u>	96
8.2.2	<u>Comparison of application server performance with respect to XML vs. non-XML type messages.</u>	97
8.2.3	<u>Analysis of time taken to process multiple XML messages</u>	98
<u>Chapter 9</u>	<u>: CONCLUSION</u>	99
<u>Chapter 10</u>	<u>: REFERENCES</u>	102
<u>Addendum A: IGUANA Structured Query Language Daemon (ISQLD)</u>		104
<u>Addendum B: IGUANA LDAP Schema</u>		106
<u>Contact Information</u>		108

List of figures

<u>Figure 1: Traditional architecture requiring separate client applications</u>	2
<u>Figure 2: Proposed architecture depicting protocol-to-XML clients interfacing to an XML gateway</u>	3
<u>Figure 3: Entries, Attributes and Values [1]</u>	12
<u>Figure 4: Database table structure [2]</u>	14
<u>Figure 5 : XML data, the parser, and the application [6]</u>	26
<u>Figure 6: Overview of Web Services</u>	37
<u>Figure 7: Current System Overview</u>	41
<u>Figure 8: Overview of common messaging system using XML</u>	43
<u>Figure 9: Functional block diagram of the proposed system</u>	47
<u>Figure 10: Application Architecture</u>	60
<u>Figure 11: XML processing for browser</u>	62
<u>Figure 12: Representation of the business layer</u>	62
<u>Figure 13: External Interfaces</u>	63
<u>Figure 14: Fieldbus package diagram</u>	66
<u>Figure 15: FieldBusBean class diagram</u>	66
<u>Figure 16: Sending a command from the front end</u>	67
<u>Figure 17: Translating a response into HTML</u>	69
<u>Figure 18: Snapshot of user interface for DPINFO command</u>	70
<u>Figure 19: Gateway package diagram</u>	71
<u>Figure 20: ParseXmlToProtocolCommands class diagram</u>	72
<u>Figure 21: Parsing and processing XML data</u>	74
<u>Figure 22: ProcessEsdProtocol class diagram</u>	76
<u>Figure 23: Connecting to ESD server and processing commands</u>	77
<u>Figure 24: ProcessSqlProtocol class diagram</u>	78
<u>Figure 25: Connecting to MySQL database server and processing commands</u>	79
<u>Figure 26: ProcessLdapProtocol class diagram</u>	80
<u>Figure 27: Connecting to openLDAP server and processing commands</u>	81
<u>Figure 28: esd package diagram</u>	82
<u>Figure 29: EsdCommander class package</u>	83
<u>Figure 30: sql package diagram</u>	83
<u>Figure 31: SqlCommander class diagram</u>	84
<u>Figure 32: ldap package diagram</u>	84
<u>Figure 33: LdapCommander class diagram</u>	85
<u>Figure 34: library package diagram</u>	85
<u>Figure 35: Application flowchart</u>	87
<u>Figure 36: Request and single command sent multiple times</u>	93
<u>Figure 37: Request with multiple commands sent once off</u>	94
<u>Figure 38: Graphical representation of number of XML messages and time</u>	95
<u>Figure 39: Graphical representation of time taken to process an XML command as the number of commands increase</u>	96

Table of tables

<u>Table 1: Database field types</u>	15
<u>Table 2: Differences between directories and databases</u>	17
<u>Table 3: ESD request-response commands</u>	24
<u>Table 4: DTD Syntax</u>	29
<u>Table 5: Debug levels</u>	88
<u>Table 6: No XML vs. XML for ESD</u>	90
<u>Table 7: No XML vs. XML for MySQL</u>	90
<u>Table 8: No XML vs. XML for openLDAP</u>	91
<u>Table 9: Time difference for application servers without XML</u>	92
<u>Table 10: Time difference for application servers with XML</u>	92
<u>Table 11: Time taken to process Single and Multiple command XML messages</u>	95