A Framework for Multimedia Collaborative Spaces

By

Maher Sobhi Adi

A thesis submitted to the faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of

Master of Applied Science

Ottawa-Carleton Institute for Electrical and Computer Engineering
Faculty of Engineering
Department of Systems and Computer Engineering

Carleton University
Ottawa, Canada
September 2006

© Copyright 2006, Maher Sobhi Adi
NOTICE:  
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author’s permission.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

AVIS:  
L’auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L’auteur conserve la propriété du droit d’auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n’y aura aucun contenu manquant.
The undersigned recommend to
The Faculty of Graduate Studies and Research
acceptance of the thesis,

A Framework for Multimedia Collaborative Spaces

Submitted by
Maher Sobhi Adi
In partial fulfillment of the requirements for the degree of

Master of Applied Science

Professor David C. Coll, Thesis Supervisor

Chairman, Department of Systems and Computer Engineering

Carleton University
Ottawa, Canada
September 2006
Abstract

On-line multimedia collaboration is a subject of great interest for people involved in globalization and education. For that end, the objective of this research is the implementation of the rather abstract idea that multimedia collaborative spaces, such as an on-line learning environment, can be constructed using Object-Oriented and Service-Oriented Architectures, and the tools and services that are provided by today's communication networks. The challenge was to create a method for using those architectures and tools.

The thesis presents, as a solution, a generic, expandable, flexible, interoperable and implementable framework for building multimedia collaborative spaces, including the management of multimedia contents. The validity of the solution is then demonstrated by the design and implementation of the illustrative example of a TeleLearning application.

The seven chapters that make up the thesis start with an introduction to the hypothesis and problems to be solved in order to demonstrate its validity. This is followed by a literature review pertaining to the thesis background, motive, and context. The technological context that led to the thesis hypothesis is described; followed by an analysis of the thesis problem; the proposed solution; a proof of concept, and ends with discussion of the results, limitations, and suggested future work.
Acknowledgement

I find it hard to express my feeling of gratitude and thanking to a great person that I have known for over six years, my thesis supervisor Dr. David C. Coll. I am trying my best to avoid the clichés when describing what Prof. Coll has provided me over the years of my research. Even though I am certain that Prof Coll does not need my thanking as a testimony to his excellent supervision and guidance, I owe it to thank him by the conviction of my religion of Islam that teaches me: “Those who thank not people, thank not God”.

Dr. Coll’s unabated and phenomenal support went beyond the formal call of duty to be the one person who saved the day and made it possible for me to continue in this overdue research work. I wish one day I will be able to reciprocate his favours. I pray to God that He take care of him and his family and maintain his life full of health, joy and happiness.

I can never forget to thank my respected parents, dear wife Nor Hayati, and beloved children, Haleema, Hanan, Hussein and Hashem, whom for years had to put up with the clutter on my desk. I thank them for their constant encouragement and support.

Finally, I seize this moment to thank my friends and fellow graduate colleagues for their time and for sharing their technical experience and insight.

To everyone, I say once again, Thank You!
Table of Contents

Abstract ............................................................................................................................. iii
Acknowledgement ........................................................................................................... iv
Table of Contents ............................................................................................................. v
List of Figures ................................................................................................................ xiii
List of Tables .................................................................................................................. xv
Acronyms ........................................................................................................................ xvi

Chapter 1: Statement of Research Problem ................................................................. 1
  1.1 Thesis Statement ................................................................................................. 1
  1.2 Thesis Hypothesis ............................................................................................... 2
  1.3 Domains of Technology Advancement ............................................................ 3
    1.3.1 Advancement on the End-User Front .......................................................... 3
  1.4 Advancement on the Enterprise Front .............................................................. 3
  1.5 Advancement on the Technological Application and Practice ....................... 3
  1.6 Impact on the Field of Online Collaboration and Distance Learning ............. 4
  1.7 Thesis Goal ......................................................................................................... 5
  1.8 Thesis Approach ................................................................................................. 6

Chapter 2: Literature Review ......................................................................................... 8
  2.1 Management of Multimedia Contents ............................................................... 8
  2.2 Frameworks for the Implementation of Collaborative Spaces ....................... 10
  2.3 Current Collaboration Tools ............................................................................... 13
    2.3.1 WebCT ........................................................................................................ 14
    2.3.2 Windows Office Communicator (WOC) 2005 ........................................... 17
Chapter 4: Development of the Basic Assumptions

4.1 Introduction

4.2 Research Statement Detailed Analysis

4.3 The Current State of the Art in Information Technology

4.4 The Tools, Services, Infrastructure, Standards, Best Practices

4.4.1 Software Development Tools and Resources

4.4.2 Internet-based Services and Free Resources

4.4.3 Communication Infrastructure

4.4.4 Standards

4.4.5 Best Practices

4.5 Broadband Multimedia Collaborative Spaces (BMCS)

4.5.1 Broadband

4.5.2 Multimedia

4.6 Collaborative Space (CS)

4.6.1 Definition of Collaborative Space

4.6.2 An Open-Ended Collaborative Space

4.6.3 Comprehensive Functionality

4.6.4 De-coupled and Flexible Design

4.6.5 Extensible Infrastructure

4.6.6 Forward-Compatible Multimedia Type Handling

4.6.7 Reusable Components
4.7 Service-Oriented Architecture (SOA) Based Framework .........................71

4.7.1 Benefits of SOA ...........................................................................................73

4.7.2 Loose Coupling .............................................................................................73

4.7.3 Modular Systems ..........................................................................................74

4.7.4 Non-Intrusive ...............................................................................................74

4.7.5 Standards-Based .........................................................................................75

4.8 Utilizing SOA for Building Extensible Systems ............................................75

4.8.1 Problem with Client-Server Approach .........................................................76

4.8.2 Solution for 4.8.1 ........................................................................................76

4.8.3 Problem with Single Services Middle-Tier ..................................................77

4.8.4 Solution to 4.8.3 ............................................................................................78

4.8.5 Problem with Multi Process Service ............................................................79

4.8.6 Solution to 4.8.5 ............................................................................................79

4.9 Rundown Scenario of an SOA Extensible System .........................................80

4.10 SOA Final Notes ............................................................................................83

4.11 Conclusion .......................................................................................................84

Chapter 5: The Proposed Framework .................................................................85

5.1 Introduction .......................................................................................................85

5.2 Detailed Breakdown of the Thesis Solution .....................................................86

5.3 Multimedia Contents in Collaboration Environments .....................................87

5.4 Thesis Problem Part I: Management, Maintenance and Delivery of Multimedia
Contents ...............................................................................................................90

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
5.5 Solving Part I of Thesis Problem: Streamline the Storage, Maintenance and Management of Multimedia Contents ................................................................. 93

5.5.1 Solution for Handling Different Types of Multimedia Contents .......... 94

5.5.2 The Unified Media Type ........................................................................... 94

5.5.3 Base64 Encoding ........................................................................................ 95

5.6 Benefits of Using Base64 Encoding: ............................................................ 100

5.7 Leveraging the Relational Database Management System (RDBMS) for Maintenance and Management of Media Contents ...................................................... 101

5.8 The Challenge of Delivering Multimedia Contents ...................................... 107

5.9 Employing A Service Oriented Architecture for Delivery of Multimedia Contents .................................................................................................................. 109

5.9.1 The Service Provider ................................................................................ 111

5.9.2 The Service Consumer .............................................................................. 112

5.9.3 SOA Data Access Layer ........................................................................... 112

5.10 Thesis Problem Part II: A Generic Tool for Multimedia Collaboration ... 115


5.11.1 N-Tier Stacked Model ............................................................................. 119

5.11.2 SOA between Tiers ................................................................................. 120

5.11.3 Smart Client Presentation Layer ............................................................... 121

5.11.4 The Service Layer .................................................................................. 123

5.12 Conclusion .................................................................................................... 125

Chapter 6: Collaboration in a TeleLearning Environment ................................. 126

6.1 Introduction .................................................................................................... 126
6.2 Definition of the Virtual Classroom Environment ........................................127
6.3 Scope of Online the TeleLearning Environment Definition .........................128
6.4 Components and Players of TeleLearning Environment .........................132
6.4.1 The Learner .................................................................................................133
6.4.2 The Course Conductor ...............................................................................133
6.4.3 The Content Author ...................................................................................133
6.4.4 Administrators ............................................................................................134
6.4.5 The Course ..................................................................................................134
6.4.6 The Tools and Services ..............................................................................134
6.5 Designing the TeleLearning Environment .....................................................135
6.6 High Level Description of the Prototype .......................................................137
6.6.1 The Learning Contents, Applications, and Users Profile Manager ......139
6.6.2 Base64 Encoder Decoder (BED64) ..........................................................145
6.6.3 The Content CRUD Gateway (CCG) .......................................................148
6.6.4 Course Browser ...........................................................................................149
6.6.5 Chat Window ..............................................................................................150
6.6.6 Web Search Tool ..........................................................................................154
6.6.7 Virtual Classroom Space (VCS) ...............................................................157
6.6.8 Main Collaboration Shell (MCS)...............................................................159
6.7 Summary and Conclusion ...............................................................................163

Chapter 7: Conclusions and Future Work .........................................................165
7.1 Discussion ........................................................................................................165
7.1.1 Designing in Abstraction and Not in Isolation ............................................165

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
7.1.2 In TeleLearning Learning Objects Include the Software Tools Used ... 166

7.2 Contributions ................................................................................................... 167

7.2.1 Contributions Strengths and Opportunities ............................................ 168

7.2.2 Contributions Risks and Weaknesses .................................................... 169

7.2.3 Contribution Limitation ........................................................................... 170

7.3 Conclusion ...................................................................................................... 171

7.4 Future Work .................................................................................................... 171

References ............................................................................................................ 173
List of Figures

Figure 3.1: IPv4 Header [38] ........................................................................................................25
Figure 3.2: IPv6 Packet Schema [42] .........................................................................................27
Figure 3.3: IP-Based Internet [46] ............................................................................................29
Figure 3.4: An Intranet [48] ......................................................................................................31
Figure 3.5: Typical Wi-Fi Network ..........................................................................................32
Figure 3.6: Hyper-Threaded CPU [56] .....................................................................................35
Figure 3.7: Generic Figure of a Dual Core CPU ......................................................................36
Figure 3.8: ISO-OSI Stack .........................................................................................................41
Figure 3.9: Basic Service-Oriented Architecture .................................................................49
Figure 3.10: Web Services Roles, Operations and Artifacts [83] .............................................50
Figure 3.11: Example of a SOAP Request Message .................................................................53
Figure 3.12: Example of a SOAP Response Message ..............................................................54
Figure 3.13: Media File Extensions Audio and Image ..........................................................58
Figure 3.14: Media File Extensions Text and Video ...............................................................59
Figure 4.15: Basic SOA Model .................................................................................................72
Figure 4.16: A typical client/server interaction [99] ................................................................76
Figure 4.17: SOA Model as Replacement for Client-Server Model .......................................77
Figure 4.18: Multi-Process Service Layers .............................................................................78
Figure 4.19: Multi-Tier Service Layer Model .........................................................................80
Figure 4.20: SOA Usability Graph ..........................................................................................82
Figure 5.21: Solution- Sub Components ...............................................................................87
Figure 5.22: Simple Directory Structure ...............................................................................90

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
List of Tables

Table 5.1: Full Character Set of Base64 Encoding ......................................................95
Table 5.2: Base64 Examples.........................................................................................97
Table 5.3: Sample Profile of Base64 Object in the Database Table.........................103
Table 6.4: Real and Virtual World Experience Comparison......................................129
### Acronyms

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full Form</th>
</tr>
</thead>
<tbody>
<tr>
<td>AOP</td>
<td>Aspect Oriented Programming</td>
</tr>
<tr>
<td>API</td>
<td>Application Programming Interface</td>
</tr>
<tr>
<td>BLOB</td>
<td>Big Large Binary Object</td>
</tr>
<tr>
<td>BMCS</td>
<td>Broadband Multimedia Collaborative Space</td>
</tr>
<tr>
<td>CDMA</td>
<td>Code Division Multiple Access</td>
</tr>
<tr>
<td>CGI</td>
<td>Common Gateway Interface</td>
</tr>
<tr>
<td>CLEC</td>
<td>Competitive Local Exchange Carriers</td>
</tr>
<tr>
<td>CORBA</td>
<td>Common Object Request Broker Architecture</td>
</tr>
<tr>
<td>CPU</td>
<td>Central Processing Unit</td>
</tr>
<tr>
<td>CRUD</td>
<td>Create, Retrieve, Update, Delete</td>
</tr>
<tr>
<td>CSMA/CD</td>
<td>Carrier Sense Multiple Access with Collision Detection</td>
</tr>
<tr>
<td>DCOM</td>
<td>Distributed COM Object Model</td>
</tr>
<tr>
<td>DLL</td>
<td>Data Link Library</td>
</tr>
<tr>
<td>DSL</td>
<td>Digital Subscriber Line</td>
</tr>
<tr>
<td>DUPS</td>
<td>DLL Universal Problem Solver</td>
</tr>
<tr>
<td>FTP</td>
<td>File Transfer Protocol</td>
</tr>
<tr>
<td>HDD</td>
<td>Hard Disk Drive</td>
</tr>
<tr>
<td>HT</td>
<td>Hyper-Threading</td>
</tr>
<tr>
<td>HTML</td>
<td>Hyper Text Mark-up Language</td>
</tr>
<tr>
<td>HTTP</td>
<td>Hyper Text Transfer Protocol</td>
</tr>
<tr>
<td>IDC</td>
<td>International Data Corporation (IDC)</td>
</tr>
<tr>
<td>IETF</td>
<td>Internet Engineering Task Force</td>
</tr>
<tr>
<td>Acronym</td>
<td>Definition</td>
</tr>
<tr>
<td>---------</td>
<td>------------</td>
</tr>
<tr>
<td>IM</td>
<td>Instant Messaging</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>IPv6</td>
<td>Internet Protocol version 6</td>
</tr>
<tr>
<td>IRC</td>
<td>Internet Relayed Chat</td>
</tr>
<tr>
<td>ISO-OSI</td>
<td>International Standard Organization's Open System Interconnect</td>
</tr>
<tr>
<td>ISP</td>
<td>Internet Service Provider</td>
</tr>
<tr>
<td>ISV</td>
<td>Independent Software Vendor</td>
</tr>
<tr>
<td>JVM</td>
<td>Java Virtual Machine</td>
</tr>
<tr>
<td>LMS</td>
<td>Learning Management Systems</td>
</tr>
<tr>
<td>LOM</td>
<td>Learning Object Metadata</td>
</tr>
<tr>
<td>MCE</td>
<td>MultiMedia Collaborative Environment</td>
</tr>
<tr>
<td>MIME</td>
<td>Multipurpose Internet Mail Extensions</td>
</tr>
<tr>
<td>MSDN</td>
<td>Microsoft Developers Network</td>
</tr>
<tr>
<td>NAT</td>
<td>Network Access Translation</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>OBJ</td>
<td>Ottawa Business Journal</td>
</tr>
<tr>
<td>OOP</td>
<td>Object Oriented Programming</td>
</tr>
<tr>
<td>ORB</td>
<td>Object Request Broker</td>
</tr>
<tr>
<td>PDF</td>
<td>Portable Document Format</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
</tr>
<tr>
<td>RAM</td>
<td>Random Access Memory</td>
</tr>
<tr>
<td>RDBMS</td>
<td>Relational Database Management Systems</td>
</tr>
<tr>
<td>ROI</td>
<td>Return on Investment</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Full Form</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
</tr>
<tr>
<td>RPC</td>
<td>Remote Procedure Call</td>
</tr>
<tr>
<td>SCORM</td>
<td>Sharable Courseware Object Reference Model</td>
</tr>
<tr>
<td>SGML</td>
<td>Standard Generalized Mark-up Language</td>
</tr>
<tr>
<td>SLA</td>
<td>Service Level Agreement</td>
</tr>
<tr>
<td>SMT</td>
<td>Simultaneous Multi-Threading</td>
</tr>
<tr>
<td>SOA</td>
<td>Service-Oriented Architecture</td>
</tr>
<tr>
<td>SOAP</td>
<td>Simple Object Access Protocol</td>
</tr>
<tr>
<td>SOC</td>
<td>Service-Oriented Computing</td>
</tr>
<tr>
<td>SODA</td>
<td>Service Oriented Database Architecture</td>
</tr>
<tr>
<td>SQL</td>
<td>Structured Query Language</td>
</tr>
<tr>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>TDD</td>
<td>Test Driven Development</td>
</tr>
<tr>
<td>UCS</td>
<td>Universal Character Set</td>
</tr>
<tr>
<td>UDDI</td>
<td>Universal Description, Discovery, and Integration</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>VCE</td>
<td>Virtual Classroom Environment</td>
</tr>
<tr>
<td>W3C</td>
<td>World Wide Web Consortium</td>
</tr>
<tr>
<td>WiFi</td>
<td>Wireless Fidelity</td>
</tr>
<tr>
<td>WOC</td>
<td>Windows Office Communicator</td>
</tr>
<tr>
<td>WS</td>
<td>Web Services</td>
</tr>
<tr>
<td>WSDL</td>
<td>Web Services Description Language</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Mark-up Language</td>
</tr>
<tr>
<td>XP</td>
<td>Extreme Programming</td>
</tr>
</tbody>
</table>
Chapter 1: Statement of Research Problem

In this chapter we will present the thesis statement, which offers a one-sentence summary of what the thesis is dealing with. Following that, we present the hypothesis that justified this thesis work. We also touch briefly the topic of advancement in technologies that created trends, and what domain they affected and in what form. After that, we discuss the thesis goals, which are what constitute the thesis problem, as solving or realizing these goals, would mean the thesis problem has been solved. The chapter concludes with presenting the experimental approach that the thesis will follow in its attempt to solve the thesis problem.

1.1 Thesis Statement

This research is founded on the premise\(^1\) [1] that the current state of the art in the field of information technology provides all the tools, services, infrastructure, standards, best practices and science that allows us to build multimedia collaborative spaces made up of reusable components based on Service-Oriented Architecture, that are open-ended, comprehensive, extensible and forward-compatible.

\(^1\) "The dominant message is that current trends in network development will provide network services that will allow the creation of compete, interactive, interoperable distance education systems into the learning process." Coll, D., "Trends in Telecommunications and Their Impact on TeleLearning", Department of Systems and Computer Engineering Carleton University, Ottawa, ON. A Report Submitted to the TeleLearning Network of Centres of Excellence, March 21, 2000.
1.2 Thesis Hypothesis

The conjecture we are examining is that today’s technology and networks have matured to provide sophisticated tools, services, standards, best practices, methodologies and trends and that these provisions constitute the context that can be utilized to build enterprise-scale collaboration environments that are standardized and forward compatible. We further postulate that to achieve future-proof capability collaborative environments should be composed of reusable and replaceable components. We will demonstrate the validity of this conjecture by way of “proof-of-concept” examples based on a framework we propose and that provides for the implementation of systems with the desired characteristics mentioned above.

Established and emerging information technologies have been playing a major role in the way we conduct our daily affairs since the middle of the last century. Most recently, the Internet and its associated services and tools have assumed the driver’s seat for creation of trends and methods affecting how we communicate and collaborate. Web browsers have arguably been the most useful piece of software for the past ten years and will continue to be so for the next generation. In fact, web browsers have evolved to be an execution environment with built-in security and run-time models.

The relationship between technology and trend is cyclic. Some technology creates new trends. Some technology is created in response to trends. These new technologies in turn create new trends, which require new technologies.
1.3 Domains of Technology Advancement

We can categorize these trends based on the domain of applicability or interaction, namely, the user domain, the enterprise domain and the technology provider domain.

1.3.1 Advancement on the End-User Front

On the end-user front, communication technology and the Internet created the communication trends that we experience today. E-mail is the de facto preferred way of exchanging messages. In the year 2002, 31 billion email messages were exchanged. In that year also IDC (International Data Corporation) [2] predicted that the number of emails would double to 61 billion messages [3]. In December 2005 IDC published IDC’s seventh annual study [4] called: “Worldwide Email Usage 2005-2009 Forecast,” which predicts that “nearly 84 billion emails (compared to 61 billion predicted in 2002) will be sent daily worldwide in 2006” [5]

1.4 Advancement on the Enterprise Front

On the enterprise front information technology is making tapping into the omnipresent and free Internet possible, affordable and seamless.

1.5 Advancement on the Technological Application and Practice

The trend in the technology sector the information technology would be to produce and provide the tools (web browser, broadband wireless networks, bigger RAM, faster CPU etc), standards (XML, UML, XAML, LOM etc) and interfaces (Web Services, COM+, etc).
WIFI) to integrate various technological pieces using the Internet as the underlying infrastructure. It is important to highlight the impact of Web Services, in standardizing the Internet-enabled interfaces between web-based resources in a service consumer-provider paradigm. Web Services open the door to new realities and opportunities that were just dreams in the past century.

1.6 Impact on the Field of Online Collaboration and Distance Learning

The field of distance collaboration is a major beneficiary of these new technologies and trends. Never before has the world of science and knowledge been so accessible and connected. With 100's of satellites in outer space and millions of miles of landline communication networks spanning the globe, terabytes of data get exchanged hourly between fixed and mobile digital terminals. With the sophisticated and robust communication infrastructure in place, distance and time zones vanish and the pressing issue becomes finding the right mix of technology to realize real online collaborative environments.

In a 1999 report to the TeleLearning Network Centre of Excellence, Prof. David Coll conjectured that “The dominant message is that current trends in network development will provide network services that will allow the creation of compete, interactive, interoperable distance education systems into the learning process.” with the advice that “the TL-NCE should aim its research towards a horizon of a richer, more useable, distributed, intelligent, broadband, multimedia, communications environment.”
This thesis provides a report on a study of the current state of the art regarding applicable “network services” to determine whether or not the original conjecture is still applicable and, if so, to propose a framework for the implementation of open-ended, comprehensive, extensible and forward-compatible systems. The study has led to the development of an implementation framework for interactive collaborative spaces based primarily on services that arisen in the intervening years.

The thesis describes a set of characteristics that such a framework should have and the services that could be used to build it. The proposed framework is component-based and uses Service-Oriented-Architecture (SOA) for inter-component communications, together with a relational database for the storage of data, contents and processes alike in a common format. The utility of the design concept is then illustrated with a demonstrative example of a TeleLearning TeleTeaching application.

1.7 Thesis Goal

The thesis hence aims at achieving the following main goals:

1. Find a generic definition for interactive multi-media collaborative spaces,

2. Present a simplified generic forward-compatible framework for managing multimedia contents,

3. Present a simplified generic forward-compatible framework for building multi-media collaborative systems,
4. Define the various reusable components of multi-media collaboration in the context of a TeleLearning environment,

5. Identify a standardized communication protocol between these components,

6. Develop a working model of a multimedia collaborative space to demonstrate that the goals have been achieved.

1.8 Thesis Approach

The thesis will follow an experimental, investigative, analytical and illustrative implementation-based approach, which will:

1. Investigate the state of the art with dealing with subjects:
   a. The proliferation of online TeleLearning systems
   b. The technologies and tools currently in use
   c. The availability of standards and protocols

2. To analyse and identify
   a. The current and predicted challenges
   b. The different aspects of the system lifecycle

3. Propose a complete framework and model of a comprehensive component-based system, with reusable/replaceable components that capitalizes on the technologies currently available.
4. Apply the findings into the implementation of a working model of multimedia collaborative space for a TeleLearning /TeleTeaching environment.
Chapter 2: Literature Review

This thesis is implementation-geared rather than theoretical; hence the focus of the literature review is on implementable systems that support the management of multimedia contents. We also review literature that deals with implementable frameworks for building multimedia e-learning spaces to capitalize on the findings of other researchers. Lastly we review standards and their impact on new software and frameworks developed.

The literature review is organized into six sections as follows:

- Section 2.1 reviews literature on management of multimedia contents.
- Section 2.2 reviews literature about development of generic frameworks for the implementation of collaborative spaces.
- Section 2.3 reviews literature about current collaboration tools.
- Section 2.4 reviews literature about standards relating to collaborative spaces.
- Section 2.5 reviews literature about technology adoption.
- Section 2.6 provides a summary of lessons learnt from the literature review.

2.1 Management of Multimedia Contents

The various servers handled the various expected processes involved in the storage, management and retrieval of multimedia contents. Their architecture was an expansion of a previous work (2004) [7] in which they aimed to “refine and extend the architecture, and to make it as general as possible so as not to be restricted to a specific standard, since our first version was too focused on MPEG-21”. The MPEG-21 standard is divided into several parts, which deal with different aspects of multimedia information management. In the MPEG-21 context, the information is structured in Digital Items, which are the fundamental unit of distribution and transaction. Digital items are digital documents written in XML according to a XML Schema.

Yoshitaka, A and Ichikawa, T. (1998) [8] highlighted the fact that conventional database systems are designed for managing textual and numerical data, and retrieving such data is often based on simple comparisons of text/numerical values. However this simple method of retrieval is no longer adequate for multimedia data, since the digitized representation of images, video, or data itself does not convey the reality of these media items. Hence integrating metadata information about each of these multimedia contents would establish an innovative way of storing and retrieving multimedia contents from systems that are conventionally not designed for such scenarios.

Yoshitaka, Kishida, Hirakawa and Ichikawa (1994) [9] proposed a system for querying and content-based retrieval that considers audio or visual properties of multimedia data by using a model tailored for multimedia data representation. Christian Süß (2000), [10] also expanded on previous work, presented “a meta-modeling approach to adaptive
knowledge management by introducing an application-specific layer.” The objective is to allow description of the conceptual document structure of knowledge modules in an application-specific way, which is needed in composition, navigation and adaptation.

Mazurek, Stroinski and Szuber (1998) [11] noted that the universal nature of the digital library makes it an “excellent foundation for other multimedia-based services such as distance learning”. They moved on to describe their first attempts to build a digital library within their native Poland as a base system supporting different application areas. In their implemented prototype, Mazurek, Stroinski and Szuber employed different types of servers serving specific types of objects can be used to store objects:

- Oracle Video Server for storing digital video objects,
- Progressive Networks RealAudio Server for storing audio objects,
- Oracle Web Application Server for storing all others file based objects.

### 2.2 Frameworks for the Implementation of Collaborative Spaces

This section reviews some of the proposed frameworks that lead to building robust systems for collaborating online. These proposed frameworks satisfied a set of strict requirements in order to really fill the gap between the underlying infrastructure, the communication protocols, standards and the user-interface.

Xu, Yin and El Saddik (2003a) [12] considered the question of what kind of solution for online distance learning could raise the e-learning systems into a higher state where
communication of heterogeneous learning systems can take place dynamically. In their paper the answer was to deploy a Web Service oriented framework in order to achieve scalable flexibility system-wide. In addition, they argued that in order for any generic framework to become a truly dynamic model, the framework must support open architectures, be highly interoperable for information exchange, and be flexible, re-usable, and compatible with other systems.

Liu, El-Saddik and Georganas (2003) [13] proposed a functional architecture to build standard-driven distributed and interoperable learning systems. The architecture defined the way to combine an information model with a component model and used Web Service as the communication protocol between each of these components. The proposed architecture also defined the various components that make up an e-learning environment. The implementation of the proposed framework was on J2EE platform.

Yin, Xu and El Saddik (2003b) [14] discussed the “magic of metadata” where information about the learning objects are stored and mapped to the actual learning objects themselves. The conjecture is that combining all proposed frameworks together leads to building a standard-oriented data model for the learning object, building collaboration environments that are made up of components that communicate using Web Services.

Web Service was found to solve the problem of interfacing and communication horizontally and vertically between the various components in a standardized fashion. The proposed framework is implementable as it uses mature and well-followed standards,
mainly XML. If this framework is followed, multi-tier collaborative applications that are expandable and interoperable can be built based on SOA. Hence, the common denominator for the proposed system was Service-Oriented Architecture and components-based applications.

They also noted that with Web Services emerging as a strong communication suite, more models are developed to extend the usability life of system. For example, the ability to bind dynamically to a Web Service at run-time rather than at design time is becoming increasingly important in the area is Service-Oriented Computing, or SOC.

Buhler, Starr, Schroder and Vidal (2004) [15] argued that such functionality requires agile software structures that support loosely coupled interactions of services that are found and bonded to at run-time. They proposed a “stubless” Web Service invocation and predicted that stubless web invocation will gain more ground as more and more SOC systems are deployed. Stubless Web Service invocation is an unorthodox way of integrating Web Services between communicating systems. It is certainly very innovative and more so applicable to the field of E-learning as it offers far reaching flexibly in exposing services or making change to the service definitions without affecting the invoking clients of the service.

Buhler, Starr, Schroder and Vidal [16] maintained however that more theoretical and practical work remains to be done to transform the vision of stubless Web Service invocation into reality; as change needs to occur at both the infrastructure and application levels.
2.3 Current Collaboration Tools

The world of collaboration encompasses many different forms and modes of communication, such as, voice, video, and text, in a synchronous or asynchronous mode. According to Nortel, choosing the right form of media to "create consistent collaborative experience across all types of access environments"\(^2\) is critical to the success and acceptability of the developed systems.

Collaboration can be divided into three main groups [17].

- Communication, such as email, websites, fax, voicemail, etc.
- Conferencing, such as Internet Relay Chat (IRC), Instant Messaging (IM), video, voice and data conferencing, application sharing, Internet forums, chat rooms, etc.
- Coordination and Management

Each of these groups requires specific tools to support its intended purpose. The number of available collaboration tools is enormous. However, all these tools allow at best a partial level of integration with other systems.

The systems investigated did not exhibit a high level of openness to integrate with other systems. There are many collaboration systems that can be reviewed.

To get an idea of how many seriously implemented collaboration tools are out there, Microsoft alone offers the following tools: Exchange 2003, Live Communications Server 2005, Live Meeting 2003, NetMeeting 3 (which is included in Windows 2003),


Outside of the Microsoft arena, there are numerous other tools. To name a few: WebCT, Blackboard, 24SevenOffice, Citadel/UX, eGroupWare, phpGroupWare, Simple Groupware, Global Knowledge, Desire2Learn, Moodle, Dokeos, e-College, .LRN, Edumate, Sakai Project, Claroline, Angel, LON-CAPA, Advanced Academics, and the list keeps on growing.

We selected to review Blackboard Inc. WebCT, as it is deeply involved in the academic arena, and Microsoft Office Communicator 2005, a new tool, and as such, should be more technically up-to-date.

2.3.1 WebCT

WebCT was developed in 1995 by Prof. Murray Goldberg at the University of British Columbia to prove the hypothesis that student satisfaction and academic performance could be improved through the use of a web-based educational resource.

WebCT is a software system that was meant to ease the creation of web-based learning environments and course material. In February 2006 Blackboard Inc. acquired WebCT. The WebCT name will be phased out over time in favour of the Blackboard brand [18].
Blackboard Inc. currently has two versions of WebCT: WebCT Vista and WebCT Campus Edition. Vista is a full-featured enterprise version and the Campus Edition is targeted at institutions with already available services such as file storage systems and course registration tools.

WebCT is not an open-source application and hence it cannot be expanded without Blackboard Inc. involvement. WebCT is a complex system to use and that is reflective of its richness in both features and functionality. For example, WebCT offers several ways to structure course materials, but this in itself requires extensive user training on the many available options. Hence, WebCT users ended up failing to set-up course web sites in full, leaving many unfinished sites with broken hyperlinks.

WebCT also was certainly not built as a component-based software application. In fact it is not even built using an object-oriented approach. It was not compatible with existing operating systems such as Windows in the sense that it did not allow the user to take advantage of file management and other Windows built-in GUI forms, such as printer set-up, File Save, File Save As, File Open forms. This simplistic approach to building such a huge system was due to the fact that WebCT was meant at its inception as a proof of concept and subsequently grew to become a full product.

Even though Blackboard Inc. released a new version called WebCT Vista which fixes some of the previous flaws, many of the original flaws present in the system that were
evident in previous versions remain in WebCT Vista such as being inaccessible to students with disabilities.

The primary "scripting/programming language for most of the programs and program modules in WebCT is Practical Extraction and Report Language" [19] or PERL. PERL is a single-threaded interpreted language that dominated the Common Gateway Interface (CGI) web-based applications in the early to late nineties of the past century. Single-threading and web-based applications do not mix unless we are ready to sacrifice resources and accept low performance. With single threads, every user accessing the application will invoke a full thread of the application, which is a full copy of the application. Opening the application to say a million students worldwide will require a robust server and serious hardware set-up.

To appreciate the consequence of the limitations of PERL we will talk about Java. Five main characteristics not available in native PERL are what make Java far more superior as the language of choice for the Internet. Java is a platform independent, compiled (which improves the performance substantially), object-oriented, secure, and multi-threaded software programming language.

Integrating components with PERL libraries is not easy and in many cases not possible or practical. Nowhere in the WebCT documentation does it suggest that WebCT can be integrated, in part or as a whole, with other systems.
2.3.2 Windows Office Communicator 2005 (WOC)

Released in March 2005 as a Windows Messenger successor, Office Communicator [20] is the new communications client by Microsoft. When WOC was released it “was not as comprehensive as was originally planned, but nevertheless still provides some impressive functionality” [21]. WOC integrates multimedia Instant Messaging (IM), telephony and Web conferencing capabilities into a single interface that is accessible from within Microsoft Office applications. WOC also allows connecting the PC to the telephone networks and therefore users can place calls from PC to PC by telephone. Users on WOC can see if the called party is online or offline; and can also conduct IM conversations securely with people using public IM clients like AOL, MSN, or Yahoo!

These features are certainly great and perhaps Microsoft is the only enterprise that is investing seriously in R&D resources to produce real results that can integrate with other Microsoft systems in support of collaborative spaces. All these systems seem business-oriented to support collaboration in a corporate environment.

Microsoft has yet to penetrate the academic arena at a capacity beyond the Operating Systems, Word Processing and PowerPoint Presentations. Microsoft has no suite of applications that can be called an academic-oriented application specifically tailored to support TeleLearning.

Just as in the case of WebCT, Microsoft left organizing and maintaining the multimedia objects out of their context. Both leave the onus on the end-user (the university or corporation) to decide how and where to maintain their data and what format to use. This
is a playing field that is still open for research and prototyping until an industry-wide consensus on a standard or framework comes into existence.

2.4 Standards and Collaboration

In an era of globalization, standards are a critical aspect of product development because of their inherent benefits. Shapiro and Varian (1999) [22] studied how standards make information sharing easier and increase the value of the infrastructure and consequently results in higher return on investment (ROI). Besen and Farrel (1999) [23] noted that companies benefit from standards when accessing other established systems and networks. But standards development, however, is time-consuming and requires major investment both in people, tools and costs [24].

To overcome the increased time and cost of development, standards are developed under the auspices of global organizations. These international bodies in many cases involve government financial subsidies such as is the case of the National Institute for Standards and Technology [25] or funding from a consortium of companies or interested organizations such as the case of IEEE [26], W3C [27] and other Standards bodies.

2.5 Technology Adoption

Rogers (1995) [28] defines an innovation as an idea, and an object of practice that is perceived as new. Rogers also defines diffusion as a process by which the innovation makes its way through social systems.
People and organizations are more willing to adopt new technology if it is comparable with or improves their existing values. “Compatibility with existing applications, hardware interfaces, skills and services are critical for an adopter to maintain an adequate level of user satisfaction.” [29]

Technology adoption is influenced by organizational mechanisms for communications and control, available resources and the innovativeness of the organization (Kimble and McLoughlin 1994) [30].

In an online TeleLearning setting, the perceived benefit in taking advantage of technology is the ease of use and borderless access to knowledge and information. This has great impact on the decision to adopt technology by the educational institute, the teacher and the learner. It would be unlikely for an organization to adopt new technology if the perceived benefits are not apparent or tangible (Kirby and Turner, 1993) [31].

2.6 Lessons Learnt

- On the Multimedia content management front, literature reviewed converged upon the concept of building generalized frameworks and data models for support of multimedia but none delved deeply into the actual handling of different types of content below the level identification of generic metadata. This leaves the door open to envision a design for uniformity in managing different types of multimedia contents, including future multimedia types as well, when considering an implementation framework for collaborative spaces.
• The literature review revealed relevant and recent work and publications dealing with proposing and developing generic frameworks and architectures. The studies published are reflective of the amount of interest in the field of TeleLearning and online collaboration. The common theme of these publications was to propose frameworks that would support the development of flexible and reusable multimedia collaborative spaces. Most importantly, all of the publications mention standards and their role in making collaboration a realized objective beyond theories and prototypes and into real, fully functional systems.

• A generic and dynamic framework should encompass the building of a standard-oriented data model for the learning object, a component-based collaboration space, service-oriented infrastructure, and a communication and data exchange protocol using Web Services.

• Regarding collaboration tools, time and time again, it was shown that real products on the market, both open-source and proprietary, are neither comprehensively developed to cover all aspects of real multimedia collaboration, nor are these products flexible enough to integrate with other systems. They come in a take-all or leave-all package, which seldom makes it possible to tap into or integrate the huge number of tools already developed. Building an abstraction tier to interface with these tools or re-engineer them into services accessible through service-call service-response protocols would be a major undertaking and involves massive amount of design and software code re-factoring.
On the standards side, TeleLearning and communication standards, especially MPEG-21, SCORM (Sharable Content Object Reference Model), XML and Web Services standards are mature, robust and proliferating. SCORM is an essential standard as it addresses the process of learning in detail including assessment, student tracking and feedback as important elements in the teaching and learning process [32]. These standards enable and simplify the implementation of flexible and standardized systems that exhibit a common theme of using extensive metadata to substantially enhance retrieval of the contents. Metadata in turn enhances the implemented system interoperability factor, resulting in more dynamically expansible and reusable systems.

The review also found a consensus that standards and best practice cannot be overlooked when designing major data systems that support collaboration, as ignoring standards poses risks on the success of the application and the organization itself. The intelligence collected and exchanged within the borders of a company or educational institution can create opportunities if proper best practice and standards are followed. This would create the proper dynamics for interfacing and communicating with external systems such as in a business-to-business topology or multimedia digital libraries settings, etc.

On the technology adoption side, information technology is making a phenomenal impact on businesses and lifestyle. In the era of information revolution where the world is “flattened” as Thomas Friedman [33] noted, and borders have vanished,
one cannot exclude the Internet-factor and globalization factor from the daily life of average citizens.

The next chapter contains a detailed overview of technologies that constitute the context within which the thesis hypothesis is developed.
Chapter 3: The Context: Emerging Technologies

This chapter presents a scan of the information technology landscape, which constitutes the context in which the thesis hypotheses emerged.

3.1 Introduction

This thesis was developed in a context of abundant and tremendous software and hardware and bandwidth resources that are available but under-utilized. Over the past decade technological advancement occurred over the whole spectrum of Information Technology (IT) systems; be it hardware, software or the network that is connecting them. The process of producing these technological components itself is advancing also with new standards and protocols.

For example, software Patterns and Practices [34] Baseline Architecture Toolkit (BAT), Microsoft Enterprise Library are a collection of carefully selected, designed, and integrated open source guidance assets proposed by Microsoft and others to build robust software in a robust way [35]. These new tools, services, protocols and standards support the creation of high quality, feature-rich software suites.

This broad view of the context can be granulated into categories to give a more detailed view of the environment within which information technology is thriving and advancing. It is a major undertaking to try to draw imaginary lines between the different components
that make up the information technology spectrum. The relationship between the various components is not planar but extensively interrelated.

A good example of such a relationship is the IP-based Network and its various components. A typical network is made up of four main parts:

1. **Hardware**: such as network cards, cell phones, computers, switches, etc. The hardware depends on software that makes up the operating system.

2. **Software**: such as drivers and firmware, operating systems, etc. The software depends on the hardware and services available through the network and capitalizes on the power and resources of the hardware.

3. **Standards**: such as CDMA, OSI-ISO, CSMA/CD, etc. Various parts of the network and even pieces of the same hardware communicate and interact based on predefined standardized protocols or follow a predictable scheme. Without such standards the system behaviour will be unpredictable and totally unreliable.

4. **Medium**: wireless and wired networks interface and complement each other through the various physical mediums such as coaxial, electromagnetic wave, infrared, fibre optics, etc.
When combined the whole system becomes as one unit that constitutes information technology services.

### 3.2 Broadband Networks

There are about 180 million persons, or about 3% of the world population, who are broadband subscribers around the globe as of mid-2005 with a steady growth momentum being sustained from year to year [36]. According to Internet World Stats one billion or 15.7% of the world population have Internet access [37]. If we merge the two statistics together, we find that about 18% of the people with Internet access are broadband users. Increased deployment of broadband access services will act as a driver for Independent Software Vendors (ISV) and Internet Services Providers (ISP) to implement productivity tools to empower the end-user. Generally speaking, networks can be categorized as intranets, extranets and the Internet. The bandwidth on these networks is high but utilization is visibly low. These networks can be wired, wireless or a combination of both.

![IPv4 Header](Figure 3.1: IPv4 Header [38])

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Internet, intranet and Extranets share a common protocol named the Internet Protocol (IP) which is a connectionless, simplified but powerful protocol that routes messages from the source to destination on the guaranteed premise that all connected devices have a globally unique IP address or a unique pair of numbers (such as IP number + Port number for shared connections). Figure 3.1 shows the schema of an IPv4 packet.

Networking technologies are advancing at the same rate as that of the software and hardware. Running out of unique IPv4 addresses (IPv4 supports 4.3 billion unique addresses) was imminent until Network Addressing Translation NAT [39] which rewrites and maps the external unique IP address to a local IP behind the firewall to enable multiple private hosts to share the connection using one single IP address was introduced.

In the mean time, the work on IPv6 [40] continued until the Internet Engineering Task Force adopted it in 1994. IPv6 is a more orthodox solution to the IPv4 address problem as it expands the IPv6 address to $3.4 \times 10^{38}$ addresses. Figure 3.2 shows IPv6 packet schema.

WiFi technology is proliferating in personal and professional settings. “In a survey of 2,500 users of mobile devices, IDC found that 34 percent of those who use WiFi technology connect to a wireless network at home, compared with 27 percent at work” [41]
With wireless routers speeds up to 108Mbps, which is faster than RJ 45 Ethernet cables, increasing WiFi usage would make wire line networks optimal only for long haul carriers. At the rate at which WiFi and wireless access is increasing that advantage might even soon change.

### 3.3 IP Networks

IP networks are the most prolific type of networks. IP is the delivery protocol which enables heterogeneous networks of different types, sizes and purposes to virtually fuse together to create mega networks such as the Internet.

IP is a connectionless, (which means no session state is kept within the packets transferred between the connecting terminals), and packet-switched protocol developed in
the 1970s and became the standard for almost all types of networks big and small. It does not guarantee delivery or sequential arrival of the packets and this makes it easier to implement. IP-based networks can be predominantly classified into three main groups: Intranets, Extranets and the most important one is Internet.

3.3.1 Extranet

An Extranet is a private computer network that uses Internet protocols, network connectivity, and possibly the public telecommunication system to securely share part of a business's information or operations with suppliers, vendors, partners, customers or other businesses [43].

3.3.2 The Internet

The Internet, or simply the Net, is the publicly accessible worldwide system of interconnected computer networks that transmit data by packet switching using a standardized Internet Protocol (IP) [44]. Between end-to-end connecting terminals, the Internet, as Figure 3.3 shows, is a layered network of heterogeneous but compatible layers that interface with each other in a standardized way very similar to that of the OSI-ISO stack. Figure 3.3 shows the OSI-OSI stack in the Internet context, where layers interact and interface with each other. The Internet is the most important technological component in information technology spectrum today. According to Internet World Stats [45], one billion or 15.7% of the world population have Internet access. That would justify the trend in which software is now being built around or for the Internet.
Layered Internet

Example of packet processing:
From source end node, relay node (router) to destination end node

Figure 3.3: IP-Based Internet [46]

The software makes use of the omnipresent, free global network of networks to add value to any organization, optimize its performance and make the organization and its services close to its customers and partners. The bandwidth and services that are available through the Internet are enormous but are yet to be tapped in a way befitting of the Internet potentials.

The advent of the Internet and multimedia broadband networks brought about a shift in the way software is designed and made available. Software has become within the reach of a global and progressively connected and — as Thomas Friedman notes “flattened” world. “Clearly, it’s now possible for more people than ever to collaborate and compete in real time with more other people on more different kinds of work from more different corners of the planet and on a more equal footing than any previous time in the history of
the world – using computers, e-mail, networks, conferencing, and dynamic new software” [33].

Java programming gave a totally new meaning to interoperability of software. Java coined the phrase *Platform Independent Application Software*. Microsoft then followed the footsteps of Sun Microsystems and introduced its own Windows-based platform independent framework, the .NET framework.

### 3.3.3 Intranet:

Intranets are private computer networks that use Internet Protocols (IP), network connectivity, and possibly the public telecommunication system to securely share part of an organization’s information or operations with its employees [47]. These internal networks as Figure 3.4 shows are located behind the firewall of the organization or use security protocols when using a public link in order to maintain the same level of point-to-point security.
3.4 Bandwidth to the End User

The amount of bandwidth available through networks is meeting and in some areas exceeding the current demand. Software and services that would make use of these available resources ought to be created. In multimedia collaboration, broadband network access is a requirement as the amount of data exchanged is substantial.

3.5 Wireless Network (WiFi) Proliferation

The notion of being wire-free is appealing to all the clients of Information Technology. Wireless connections have been around for over two decades when price was the main impediment to mass adoption. The past five years have seen more wireless devices sitting on desktops such as wireless mice, keyboards, routers, modems, etc. Thanks to remarkable drops in semi-conductor prices due to globalization and outsource/off-
shoring, wireless capability can be added to these devices is feasible without much effect on the price. Figure 3.5 is a graph of typical WiFi set-up. The wireless PCs connects a “hotspot” (wireless routers) using WiFi frequencies; the wireless access point is connected to the Internet through a local LAN or DSL connection.

![Figure 3.5: Typical Wi-Fi Network](image)

NetGear [49] CEO Patrick Lo justifies “this proliferation of wire-free world over the last 18 months is due to decreases in semi-conductor prices. He also continues to predict “the trend of integrating WiFi into devices is going to be accelerated and those newer digital devices such as high-definition digital TVs will come with WiFi”. We are also starting to see Internet radios, which are digital devices that not only can receive broadcast channels, but can also receive digital channels from the Internet. This is a new generation of digital devices, which will come with built-in WiFi [50].
3.6 Hardware: Increasing Capabilities Decreasing Costs

On the computer hardware front, quantum leaps in terms of power, capacity and total ownership cost for both business and home users have taken place.

3.6.1 CPU Processing Power

Processing speed is increasing at an unprecedented rate. PCs with 3.4 GHz CPUs are readily available household items. This has enabled users to have more CPU-intensive software running concurrently and to achieve results faster. New technologies such as Hyper-Threading and multi-core also increase the processing power of the CPU.

Pat Gelsinger, co-leader of Intel's Digital Enterprise Group, said that we would see 10X performance improvements over the next four years, compared to 3X in the last four years, driven by multiple cores and parallelism [51]. Martin Reynolds, Vice president of Gartner Group [51] claims that by mid-2006, the industry will experience the biggest single performance increase ever for many applications. "Word and Excel don't matter—but for video editing, dealing with Acrobat documents, simulations and applications with a workstation heritage, the performance boost is significant." [52]

3.6.2 Random Access and Archival Memory Units

Random Access Memory (RAM) and Hard Disk Drive (HDD) capacity has also advanced substantially as part of the hardware progression. RAM and HDD and other semiconductor and archival technologies are “doubling the speed of their devices every 18 months” [53]. RAM sizes play a major role in PCs processing power. Access to data cached in RAM is much faster than accessing data from a HDD. Smaller RAM forces the
operating system to rely on the virtual RAM by caching data pages on the hard disk. With many applications running, these pages increase in number and a good percentage of the CPU processing will be wasted thrashing and swapping pages in and out of the physical RAM. Bigger RAM gives the CPU the much-needed access to random and fast memory locations, which improves the overall performance of the system and also adds to the richness and quality experience of the user of a computer system.

3.6.3 Hyper-Threading (HT) and Multi-Core Technologies

The added RAM and CPU power and capacity brought parallel processing to home computers. To improve performance in the past, threading was enabled in the software by splitting instructions into multiple streams so that multiple processors could act upon them. Simultaneous Multithreading (SMT) or what has come to be known as Hyper-Threading technology “provides thread-level parallelism on each processor, resulting in more efficient use of processor resources, higher processing throughput, and improved performance on today's multi-threaded software.” [54] Hyper-Threading was first introduced on Intel’s Pentium 4 CPU; computer users have access to two virtual CPUs on one physical CPU core. This technology is particularly beneficial to multimedia applications as “Multimedia applications are especially conducive to thread-level parallelism because many of their operations can run in parallel.”[55]

Figure 3.6 shows what a Hyper-Threading CPU execution model look like. Two logical CPUs are available for the execution queue. Hyper-Threading works by duplicating
certain sections of the processor to the host operating system, allowing the operating system to schedule two threads or processes simultaneously.

Figure 3.6: Hyper-Threaded CPU [56]

On the other hand, the Multi-Core CPU technology "includes two or more complete execution cores per physical processor, enabling server platforms to handle more tasks, software threads or applications simultaneously." Combining Multi-Threading with Multi-Core provides real parallelism in the execution for multithreaded software. As shown in Figure 3.7, a dual core CPU is actually two CPUs burnt on the same chip. Multi-core chip architecture gives a real multithreaded execution environment.
3.7 Decreasing Total Cost of Ownership

Bigger RAM and faster CPUs no longer mean high prices. The price of a typically configured PC or laptop computer has dipped since the year 2000 [57]. Many configurations, which were prohibitively expensive ten years ago, are now affordable at a fraction of the original price.

3.8 Increasing Software Power

Increasing software power is not so much an increase in the processing power of software compared to previous versions, but rather with its increase of features and functionality. Software is currently built to efficiently utilize the available hardware and network resources. The software has become an integral and intelligent part of any successful and progressive organization. To appreciate the effect of software technology on productivity, a US Federal Reserve Report released in the year 2000 that indicate that the use of
technology and the production of IT goods since the mid-1990s has accounted for about $50 billion in productivity output every year, contributing about two-thirds of the $70 billion annual productivity gain over the same period [58].

3.9 Integrated Development Environments (IDE)
IDE, which normally consist of a source code editor, a compiler and/or interpreter, build-automation tools, and (usually) a debugger, have come a long way to enrich the experience and simplify the job of the developer. For example, Microsoft Visual Studio (VS) 2005 allows users to track software code changes and test results from within the IDE. It also comes with testing code coverage, which colour codes the part of the software that was never tested under all test cases. Code coverage enables the developer to quickly spot dead codes, or discover holes in the test case scenarios. There are many more features that would make building sophisticated software an easier task. With new MS VS 2005 IDE the developer [59]:

- Is more productive and obtains faster results
- Can build dynamic Windows, Web, mobile, and Office-based solutions
- Can communicate and collaborate more effectively within software teams
- Can ensure quality early and often throughout the development process.

3.10 Platform Independence and Runtime Environments
The benefits of applications that can only run on one machine or one specific platform are surely limited to users of that required environment. Platform independent applications, which can run on any platform, open a world of possibilities for those applications. Platform independence enables the integration and merging of disjoint and heterogeneous systems. This dynamic of execution on EVERY platform has been the dream of every organization that deals with software for decades.

And since collaboration, be it for financial benefit or otherwise, is the common platform for all organization, it cannot be fully achieved unless we achieve parallel, vertical and horizontal technology independence.

3.11 Categories of Interoperability

Interoperability between heterogeneous systems increases depending on the degree of collaboration and automation. “In the Internet economy, businesses and consumers are increasingly dependent on automated, interactive processes using real time Web-based interactions, software agents, and market innovations such as online auctions” [61]. Interoperability can manifest into two major categories namely, Horizontal or side by side, and Vertical or multi-tiered.

3.11.1 Horizontal/Parallel Interoperability

Horizontal interoperation means that two or more disjoint separated applications –or components of an application– can interoperate over distance. Implementing horizontal interoperability is more complex than vertical as more and more interfaces and protocols
will need to interact and communicate to ensure end-to-end co-operation. The horizontal model involves the utilization of an underlying network, reconfiguration of the security requirements and implementing an interface between the various components.

Parallel Interoperability has been the most conspicuous to end users. Often installing applications caused trouble to another application running on the same machine. This can happen for different reasons such as using different versions of a DLL. This problem is called DLL Hell. Anderson [61] describes three types of DLL Hell and also discusses how the DLL Universal Problem Solver (DUPS) package can be used to resolve DLL compatibility problems. Another problem that can transpire when installing or running more than one application on the same system occurs when an application locks a resource that is inherently shareable - such as a network access - and blocks other applications out. Interoperable applications such as Internet Explorer and MSN Messenger both share the network access without blocking the other from accessing the resources available.

In today's ever expanding world of computing no one can deny the benefits of full interoperability. Interoperability is a need more than a luxury. Standards and common protocols are the only way to ensure co-operation between various applications and resources. For example, common interfaces between the various layers of the ISO-OSI stack are what makes the stack the most important part of the Internet makeup. These interfaces make the various layers loosely coupled and ensure location transparency.
Technologies converge towards common platforms or a common platform that allow them to grow vertically, i.e. be able to run on different machines without the need to reconfigure or recompile; or run horizontally by becoming available for use by more users and applications side by side. In mathematics “convergence” describes “limiting behaviour of an infinite sequence or series toward some limit” [62]. This limit can be the number of unique platforms or standards that all technologies can converge into. Does that mean when technologies grow in parallel they never converge? The subtle idea is that technologies should continue their course of advancement while keeping an eye on where other streams of technology are heading to ensure that users can capitalize on each other’s best results.

Before the pure service-oriented model was introduced in 2000, the horizontal model was implemented with technologies such as Distributed Common Object Model (DCOM) and Object Request Broker (ORB) based on the CORBA model, which are object-oriented architectures based on the Remote Procedure Call (RPC) model. These modules used different sets of standards that matured into the Web Services model.

3.11.2 Vertical Interoperability

The Internet is primarily based on vertical interoperability by means of the ISO-OSI Reference Model [63], which is a “layered, abstract description for communications and computer network protocol design, developed as part of the Open Systems Interconnection initiative” [64]. With a layered model each layer communicates and interoperates with its lower layer and its above layer using a set of standards. This
"separation of concerns" equips us with the ability to connect and use each layer through the API [Application Programming Interface] that each layer makes public. Figure 3.8 [65] shows the ISO-OSI model which is a perfect example of vertical interoperability.

![OSI Model Diagram]

**Figure 3.8: ISO-OSI Stack**

Other examples of vertical interoperability are the Java and the Microsoft .NET frameworks. The concept and the technical model are streamlined and intuitive. Applications are built and compiled into a middle level language code. In the .NET model middle-level code is called ASM. In Java the middle-level code is called Byte code. This middle language code is then run on machines that have the .NET Framework or the Java Virtual Machine (JVM) installed for Microsoft NET-based or java-based applications respectively.
These middle tier engines (.NET or JVM) are platform dependent; hence there is a .NET framework version for UNIX, Windows, and OS2, and PDAs, and so on. Also there is a JVM for Windows, UNIX, PDAs and so on. The middle tier engine resides on top of the operating system and below the application and provides the translation of ASM or Java byte codes to native machine-dependent instructions that the operating system understands.

3.11.3 Advantages of Vertical Interoperability

- Applications are much more robust and more secure. The middle tier security model can be adjusted/designed to allow limited/no access to specific resource on the machine such as the hard disk;
- Expanded application use;
- Applications are written once and used on any platform;
- Development productivity increases as code libraries can be used over and over, as in general, platform independent models are inherently object-oriented based systems;
- Lowers application operation deployment cost, as hardly anything other than the middle tier needs to be deployed to get the application working.

3.12 Microsoft .NET Framework

The .NET framework is not a programming language but a runtime and design framework that would execute all the .NET programs regardless of the underlying infrastructure.
3.13 Sun Microsystems Java

Java achieves platform independence by employing a middle tier that resides between the compiled java program and the operating system. This middle tier is called the Java Virtual Machine (JVM), which is a software implementation of a CPU that runs compiled Java code. Java programs are built to run on this virtual machine, allowing them to be run on any real machine that has a JVM. The virtual machine, JVM, is a platform-independent programming language that converts Java byte code into machine-dependent instructions and executes it [66]. Most programming languages compile source code directly into machine code that is designed to run on a specific microprocessor architecture and operating system, such as Windows or UNIX.

3.14 Design Patterns, Best Practices and Software Libraries:

It is evident that the advances that are taking place in IT are not limited to one aspect but rather across the board. Development in this field is producing new inventions and innovations, tools and processes for efficient ways to develop large-scale software function-rich multimedia applications. According to the Standish Group over 80% of projects are unsuccessful either because they are over budget, late, missing functions, or a combination of these problems. Moreover, 30% of software projects are so poorly executed that they are cancelled before completion. According to a 1995 estimate, in the USA alone, about $250 billion each year [67] is spent on software development. It is of extreme importance to choose the appropriate development lifecycle for each project at hand because all other activities pertaining to the success or failure of the project are derived from this process [68]. Design Patterns owe their importance to the
level of complexity software systems have evolved to. With Design Patterns, the complexity can be simplified or, at the very least, the management of the design lifecycle of such complex systems can be simplified.

3.14.1 Design Patterns

Design Patterns are defined as “general repeatable solutions to commonly-occurring problems” [69] and are “designed to assist developers with common enterprise development challenges” [70]. They are being proposed as part of a best practices approach to developing projects and are coupled with completely implemented software libraries that generalize the solution for such common challenges. A recent Design Pattern called Agile was introduced to expedite the software design process.

3.14.2 Agile Design Methods

The Agile design and development methods “attempt to minimize risk by developing software in short timeframes, called iterations” [71]. This method has been reported to work well for small teams of ten members or fewer, facing unpredictable or rapidly changing requirements. Agile methodologies prioritize adaptability to changing requirements over the project when predictability is valued by more traditional methodologies. One of the most well known documented types of Design Pattern methodology is called Extreme Programming (XP). The Object-Oriented paradigm, which promotes data and functionality encapsulation, is the de facto choice for major software products. Many familiar languages have become purely object-oriented, such as Visual Basic and C#.
3.14.3 Aspect-Oriented Programming

Aspect-Oriented Programming is a newly established software development paradigm and “has at its core the enabling of a better separation of concerns, by allowing the programmer to create cross-cutting concerns as first-class program modules” [72]. “Aspects” constitutes a modular representation of a cross-cutting concern. These cross-cutting concerns are complemented with “join points” and “semantic effects at join points” [73]. Aspect-oriented technology has many potential benefits [74]:

- Provides a way of specifying and encapsulating cross-cutting concerns in a system.
- Allows a better job of maintaining systems as they evolve
- Adds new features, in the form of concerns, to existing systems in an organized manner.
- The improvements in expressiveness and structure might allow systems to run longer, and to incrementally improve them without incurring the expense of a complete rewrite or substantial re-factoring.

3.14.4 Enterprise Library, Application Block

In an effort to support its proposed best practice Microsoft developed a library of open source application blocks to assist developers with common enterprise development challenges [75]. Application blocks are a type of guidance, provided as source code that can be used "as is," extended, or modified by developers to use on enterprise development projects. Full source code is available in two .NET languages, C# and VB,
and is made up of seven of the most commonly used blocks that all multi-tier applications use. The blocks are Caching, Cryptography, Configuration, Logging, Event Handling, Data Access and Security. The aim of the application blocks [76] is to:

- Reduce development cost by integrating the application blocks into a project to save on development time.
- Increase confidence in the application because the application blocks are well tested and have been used by thousands of developers around the world in real life applications.

### 3.14.5 Open Source Software

An Open Source Software (OSS) program comes with a license that provides the users the freedom to (i) run the program for any purpose, (ii) study and modify the program, and (iii) make changes to the original program without having to pay a royalty to those who developed the software [76]. There is a serious adoption of OSS initiatives by Microsoft [77] (Enterprise Library), Sun [78] (Java Enterprise System) and other technology main players.

### 3.15 Services in the Macroscope

Since the days of CORBA and DCOM generalized functionality was offered as service that is available for integration with other services. A service is a “function that is well-defined, self-contained, and does not depend on the context or state of other services” [79]. The novelty about a service though, is the new approach that evolved and took shape in the late 1990’s when XML and SOAP became standards. Replacing all the
different communication and invocation protocols such as Process Hooks, RPC and CGI, the standardized Web Service connected between services while maintaining implementation details hiding and platform independence.
3.16 Service Oriented Architecture (SOA) for Dynamic Solutions

A service-oriented architecture is a framework that has a robust service layer. SOA can be considered a natural evolution in systems architecture. SOA is a chemistry of best practices evolved from years of use of the architectures. An SOA provides a good framework from which to build the dynamic solutions required today [80]. “A service-oriented architecture is essentially a collection of services. These services communicate with each other. The communication can involve either simple data passing or it could involve two or more services coordinating some activity. Some means of connecting services to each other is needed” [80].

3.16.1 Service-Oriented Database Architecture (SODA)

Motivated by the need for building Service Oriented Architecture (SOA) features directly into a database engine [81], Microsoft developed the Service Oriented Database Architecture (SODA) for the Microsoft SQL Server DBMS [81]. This new pattern to develop data centric applications extends the powers and environment of the database using a set of SQL 2005 built-in features.

3.16.2 Web Services (WS)

A Web Service is a service that is accessible over the web. A service must comply with five standards, namely, XML, WSDL, SOAP, UDDI and HTTP to be considered a Web Service. In a recent development that would simplify interoperability between distributed computing systems, the World Wide Web Consortium (W3C) ratified Web Services
Addressing 1.0 as a standard [82]. Figure 3.9 shows a simple web-service topology. The service is invoked by a service-request issued by the service consumer. The service provider returns a response to the request. Figure 3.10 gives the full workflow cycle of the Web Service topology from publishing, discovery, and binding. Once the binding between the consumer and provider has been established, the service is invoked by service request and service response. Figures 3.9 and 3.10 provide an overview of the Web-Service workflow framework.

Figure 3.9: Basic Service-Oriented Architecture
"WS-Addressing provides transport-neutral mechanisms to address Web Services and messages. Specifically, this specification defines XML elements to identify Web Services endpoints and to secure end-to-end endpoint identification in messages. This specification enables messaging systems to support message transmission through networks that include processing nodes such as endpoint managers, firewalls, and gateways in a transport-neutral manner" [84].

3.16.3 Roles in Web Services Architecture

The roles in Web Service Architecture can be defined as follows [84]:

- Service Provider. From a business perspective, this is the owner of the service.
  From an architectural perspective, this is the platform that hosts access to the
• Service Requester is the software entity that requires certain functions to be satisfied. From an architectural perspective, the service requested is the application that is looking for and performs interaction with a service. The service requestor can be an Internet browser, a software program or another Web Service.

• Service Registry. This is a searchable registry of service descriptions where service providers publish their service descriptions. Service requestors find services and obtain binding information (from service descriptions) for services during development for static binding or during execution for dynamic binding. For statically bound service requestors, the service registry is an optional role in the architecture, because a service provider can send description to service requestors.

3.17 Standards

Over the past decade more and more protocols became standards. Some of the standards constituted a complimentary group that covers an end-to-end paradigm. Within these standards are the group of five that support Web Services.

3.17.1 Extensible Mark-up Language (XML)

The most important contribution to standards in the last decade was the endorsement of XML version 1.1 by the W3C in 1998. XML is a subset of Standard Generalized Mark-up Language (SGML) and is a mark-up language that may be used to describe any object.
The object can be a service, a literal, a document or anything that can be described. XML documents are text based and describe any object and the elements and properties of the object recursively. This provides a tree-like hierarchical text document from which, once parsed, the object and its associated properties and values can be reconstructed. “The fundamental unit in XML is the character, as defined by the Universal Character Set (UCS). Characters are combined in certain allowable combinations to form an XML “document” [85]. XML offers many characteristics that make it best suited for information exchange. Its strong syntax makes the documents easy to parse and read. UCS allows XML to support any human language within the Unicode groups. XML documents are pure text and hence are not limited to any proprietary formats or extensions and that also makes XML machine-independent and forward compatible as text is supported by all types of operating systems and on all platforms.

3.17.2 Web Services Description Language (WSDL)

The Web Services Description Language uses XML namespaces to “describe the public interface to the Web Service” [86].

WSDL describes how to bind with a Web Service and what fully qualified methods the Web Service exposes. Although not mandatory, it is a common practice in Web Services that all publicly exposed methods should return a value, at the very least an indication of success or failure. At design time the software developer links to the Web Service using the Web Service URL. Once the Web Service is bound, the WSDL profile gets returned from the server where the Web Service is published. This WSDL profile gives the detailed description of the Web Service public methods and their corresponding input and
output signatures and parameter data types. The designer now has local stubs that bind to an actual Web Service residing on the Web Service server. At run time the Web Service gets instantiated and a local virtual object of the Web Service is accessible with binds to the server. The consumer of the service remotely invokes Web Service methods using SOAP and receives the returned results as an XML document.

3.17.3 Simple Object Access Protocol SOAP

SOAP is an evolution of XML-RPC and forms the foundation layer of the Web Services stack. SOAP is the protocol that enables a Web Service to invoke methods remotely over the Internet. SOAP messages are envelopes made up of two parts, a header (optional and located at the top of the envelope) and the body which includes the method name and parameters values as illustrated in Figures 3.11 and 3.12 below.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <getUserProfile xmlns="http://Carleton.ca/wss/WebLoginWS">
      <studentID>120123456</studentID>
    </getUserProfile>
  </soap:Body>
</soap:Envelope>
```

**Figure 3.11: Example of a SOAP Request Message**
3.17.4 Universal Description, Discovery, and Integration (UDDI)

Universal Description Discovery and Integration (UDDI) is a Web Service registry system that uses the XML namespace and therefore is standardized and platform independent. As the name suggests, its primary role is to provide the mechanism to register the service for the providers and enable consumers to discover those services and integrate them. UDDI is a fundamental member of the Web Services Stack.

3.17.5 Hyper Text Transfer Protocol (HTTP)

Hyper Text Transfer Protocol (HTTP) is a simple yet robust transfer protocol for messages over the Internet. HTTP robustness is due to its reliance on a TCP/IP stack at
the network layer when communicating with the target resource. HTTP defines eight methods indicating the desired action to be performed on the identified resource.

- **GET**: Requests a representation of the specified resource.
- **HEAD**: Asks for the response identical to the one that would correspond to a GET request, but without the response body.
- **POST**: Submits user data (e.g. from a HTML form) to the identified resource. The data is included in the body of the request.
- **PUT**: Uploads a representation of the specified resource.
- **DELETE**: Deletes the specified resource (rarely implemented).
- **TRACE**: Echoes back the received request, so that a client can see what intermediate servers are adding or changing in the request.
- **OPTIONS**: Returns the HTTP methods that the server supports. This can be used to check the functionality of a web server.
- **CONNECT**: For use with a proxy that can change to being an SSL tunnel.
3.18 Database Technology

Database systems like all other sectors of information technology, have experienced major advances. With new features and increasing adaptation of these new features, the database system is playing a more important role in business and software processes than ever before. In the following section database growth in functionality and market share will be discussed.

3.18.1 Database Grows in Functionality and Features

Over the past five years Database Management Systems (DBMS) have advanced both in data size and system functionality. The newest versions of Oracle (Oracle10 G2) and MS SQL (SQL 2005) and IBM DB-2 are capable of storing objects such as images and XML documents in their original format while retaining their properties. That is not a trivial feat. It opens a whole new style of programming. For example, consider that in the old days a person’s address used to be a full record in a database table. Street name, unit number, city, country and zip code were all kept in separate fields. In the database systems, an address can be an object, which is persisted or preserved in the database as an XML document. Far more complex data structures can also be saved as an XML document in one field in one record in one table in the database. This type of functionality and many more were unheard of five years ago. The data types these new database systems can handle, the functionality they offer and robustness they exhibit is substantial. Tapping into these new Relational Database Management Systems (RDBMS) capabilities is becoming a mandatory design move in order to build and support scalable and intelligent software systems.
3.18.2 Database Growth and Market

The worldwide market for relational database management systems (RDBMS) grew by 9.4% to $14.6 billion in 2005, according to preliminary figures released by IDC. “These numbers reflect steady growth for the RDBMS software market, indicating a continued increase in the deployment of relational databases” [87].

3.19 Media and Media Extensions

Today’s data is the digital representation of five main categories of media files. The media file can be an image, text, audio, video or binary file.

The National Institute of Standards and Technology (NIST) lists several possible file extensions under those four categories [88]. The tables below which were compiled from the NIST website, list the best known media file formats and extensions. Readily available free players or viewers support all of these files. Figures 3.13 and 3.14 show the NIST categorization of media contents and the best-known media formats on the market.

Microsoft for example, has players and viewers for PowerPoint, Word, Excel, etc. MP3 players are found in abundance on the Internet. Video players are just as available also. Real and Microsoft and others all offer video players and video decoders free of charge to support their proprietary video formats. With these free players and viewers, the end-user can play or view media files by downloading the free players and viewers that support these media format without having to buy any new software.
<table>
<thead>
<tr>
<th>Format</th>
<th>Support</th>
<th>Format</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Audio</strong></td>
<td></td>
<td><strong>Image</strong></td>
<td></td>
</tr>
<tr>
<td>AAC</td>
<td>Decoder implementations are available from ARM, Cirrus Logic, Fraunhofer IIS, and Texas Instruments</td>
<td>Bitmap</td>
<td>Browsers, various image viewing and editing programs (freeware, shareware, and proprietary)</td>
</tr>
<tr>
<td>AC3</td>
<td>The standard in DVD audio</td>
<td>GIF</td>
<td>All browsers, most image viewing and editing programs - low to high-end</td>
</tr>
<tr>
<td>MIDI (* .mid)</td>
<td>All music composing software, most music editing software, and nearly all synthesizers, though support in computer audio programs is not as widespread</td>
<td>JPEG</td>
<td>All browsers, most image viewing and editing programs - low to high-end</td>
</tr>
<tr>
<td>MP3 (MPEG-1 layer 3)</td>
<td>Playable in portable MP3 players, or in programs like Winamp, RealPlayer or Windows Media Player; recent audio CD and DVD players</td>
<td>JPEG 2000 (* .jp2)</td>
<td>The W3C is not yet behind it, but there are a number of other projects trying to increase its support</td>
</tr>
<tr>
<td>RealAudio (*.ra, *.ram)</td>
<td>Playable through the free RealPlayer as well as other players and editors from RealNetworks that are available for purchase</td>
<td>MNG</td>
<td>No browser integration yet, but since the W3C is behind PNG it is likely not too far in the future</td>
</tr>
<tr>
<td>Waveform (*.wav)</td>
<td>Now readable on Macs (Apple's AIFF format is equivalent to WAV as it is also uncompressed); most systems come with a basic music program that supports WAVs</td>
<td>PNG</td>
<td>Varies widely among browsers and programs (list of supporting applications)</td>
</tr>
<tr>
<td>WMA (Windows Media Audio)</td>
<td>Windows Media Player; becoming increasingly widespread on the web</td>
<td>SVG</td>
<td>It is described in XML; Adobe provides a plug-in for Netscape and MSIE, but there are many programs for viewing/editing; W3C supported</td>
</tr>
<tr>
<td></td>
<td></td>
<td>TIFF</td>
<td>Supported by many applications; cannot be embedded into a webpage</td>
</tr>
</tbody>
</table>

Figure 3.13: Media File Extensions Audio and Image
<table>
<thead>
<tr>
<th>Format</th>
<th>Text Support</th>
<th>Video Format</th>
<th>Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASCII (*.txt)</td>
<td>Any text editor or word processor will open it, the most simple of which are Notepad or SimpleText</td>
<td>AVI (Audio Video Interleave)</td>
<td>Windows Media Player (now for Mac too) and RealPlayer; most widely used compressor is Cinepak, free with Video for Windows, though dozens of others are available</td>
</tr>
<tr>
<td>DOC (Microsoft Word Document)</td>
<td>Microsoft Word, but can be opened by WordPerfect and other higher-end word processing apps</td>
<td>DV and Mini-DV</td>
<td>Digital video cameras and non-linear video editing applications</td>
</tr>
<tr>
<td>EBCDIC</td>
<td>IBM OS/390 systems</td>
<td>MPEG (*.mp4, *.mp4, *.mp2, among others) (Motion Picture)</td>
<td>Numerous applications, including Windows Media Player and QuickTime</td>
</tr>
<tr>
<td>HTML (*.html *.htm) (Hypertext Markup Language)</td>
<td>All browsers support HTML, though the advanced features are implemented differently in each and provide non-standard extensions</td>
<td>QuickTime (*.mov)</td>
<td>Requires free download of QuickTime Movie Player to view; QuickTime Pro can be used for creating and editing</td>
</tr>
<tr>
<td>OEBPS (Open eBook Publication Structure)</td>
<td>Used by a variety of electronic publishing systems and reading devices</td>
<td>RealMedia and RealVideo (*.rm, *.rv)</td>
<td>RealPlayer, RealNetworks provides a number of apps for editing, streaming, capturing, and creating</td>
</tr>
<tr>
<td>RTF (Rich Text Format)</td>
<td>Can be opened, edited, and saved using MS Word</td>
<td>Flash and Shockwave (*.swf)</td>
<td>Requires free download of Macromedia Flash or Shockwave Player to view, purchase of Macromedia Flash or Director to create, though free 30-day trial versions exist</td>
</tr>
<tr>
<td>Unicode</td>
<td>NT and Win2000 systems; required by modern standards such as XML, SQL, etc.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>WPD (Word Perfect)</td>
<td>Windows only</td>
<td></td>
<td></td>
</tr>
<tr>
<td>XML (Extensible Markup Language)</td>
<td>Opera browser has most complete implementation, DocZilla has even more, but is still in the alpha phase; MSIE 5.5 and Netscape 6 support also</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.14: Media File Extensions Text and Video
3.20 Conclusion

This chapter has presented a summary of the advancement in the field of Information and its associated technologies. It was evident throughout the various section of this chapter that over the past decade, major sets of new software and hardware tools and standards that support the various aspects of multimedia collaboration have been produced. Design and Modeling tools such as Microsoft Visio [89] and IBM Rational Rose [90] give the power to have an overall system design picture. Tools for designing and normalizing database schemas such as Computer Associates Erwin Data Modeller [91] empower the systems architect with the ability to create and maintain databases, data warehouses and enterprise data models.

When an application has been designed and the database schemas identified, Design Patterns come to the rescue and guide the software architect to choosing the most suitable approach to building the system based on their process and system plans. Platform independence and implementation hiding can be adopted using Java or .NET and by applying a SOA approach for the system integration. CPU, RAM and HDD and other hardware improvements also add momentum to the ease of building comprehensive multimedia collaborative spaces. Media files are now supported by free applications. It can be assumed that even new future media file formats will also be supported by players that the end user can have access to through the Internet free of charge. These various aspects of technology are of great value, and are indeed the assets for creating, using and maintaining a comprehensive multimedia collaborative environment. Lastly, the Internet is expanding at a very encouraging rate.
Chapter 4: Development of the Basic Assumptions

This research is founded on the hypothesis that the current state of the art in the field of information technology provides all the tools, services, infrastructure, standards, best practices and science that allows us to build broadband multimedia collaborative spaces that are open-ended, comprehensive, extensible, forward-compatible, and made up of reusable component based on Service-Oriented Architecture.

4.1 Introduction

Multimedia Collaboration involves the exchange of information of different types (Video/Audio/Text/Images, Binary, and Data etc.) between the collaborating parties. The four major known categories of media types are text, audio, video and image. But also binary media type, such as compressed files or compiled code etc, is also an important media type that must be taken into consideration. Chapter 3 sections 3.12 dealt with these various media types and showed that there are many different file extensions associated with each media type. For example, an image file can be jpg, gif, bmp and so forth. File extensions determine the type of data stored in the file to distinguish between file formats [92].

The designers and architects of collaborative environments need to work not just with different types of media but also with different types of file extensions (media file data formats) within each of those media types, which adds an extra level of complexity to the
design and implementation process. The management of these different media files requires sophisticated processes to ensure data integrity and a high rate of availability. Generally speaking, these media files are distributed over an extensive and progressively fanned-out directory structure that resembles a logical grouping and mapping between these files and their learning object metadata. Maintaining such a complex directory structure is not a trivial task. In fact, the challenge presented thus far is only the tip of the iceberg. Exchanging these files with the various collaborating parties is also complicated.

A generic system or framework is needed to handle two major tasks, namely the management of the multimedia contents and the provisioning of tools and services that constitute the infrastructure support for collaboration. The lack of such a generic system results in the construction of inflexible, closed and complex collaborative tools.

The effort and resources invested and consumed in the process of developing a generic framework is offset by:

- The simplicity of the resultant design,
- The modularity of the various pieces that make up the collaborative space,
- The ability to replace, add or remove components in a seamless manner.

In the following section we will present a detailed dissection of the characteristics of the framework as identified in the research statement above. The context and limitations of each of these characteristics will also be identified.
4.2 Research Statement Detailed Analysis

The thesis statement argues that the various tools for the design, development and deployment of comprehensive collaborative spaces are readily available and that the current state of the art in information technology is sufficient to produce such multimedia environments for the purpose of collaboration.

The following sections will attempt to give a deeper look at the scope of the concepts presented in the research statement, which constitute the various characteristics of the proposed framework.

4.3 The Current State of the Art in Information Technology

As discussed in Chapter 3, the state of the art in the field of information technology has taken quantum leaps in terms of the quality and quantity of services and tools that have made introduction, mass production and distribution of software and services fast and reliable. All domains of information technology have undergone tremendous change. We have seen change in the hardware and software tools and services. Broadband networks have proliferated both in the wired domain and the wireless. The number of Internet users and users who have home PCs is expanding. This current state of the art makes it necessary that technology leaders streamline and standardize the ways to tap into public domain resources and especially the Internet.

4.4 The Tools, Services, Infrastructure, Standards, Best Practices
These are five fields of Information Technology (IT) identified as the requirements for building sophisticated software systems. The following sections will present and give specific examples of those requirements that are ready as of today to be part of a collaborative environment, or be part of the building process of a collaborative environment.

4.4.1 Software Development Tools and Resources

Software development tools have evolved to become comprehensive systems. Integrated Development Environments (IDE’s) provide the developer now with more than just a text editor for writing code and a compiler to compile the code. Some IDE’s, such as Visual Studio (VS) 2005 Team System Edition, come integrated with tools and functionality for real-time reporting on tasks and bug fixes in one suite that allows project management to see up-to-the-second reports on project progress, tasks assigned, code testing and bug fixes etc.

Microsoft VS 2005 was released in November 2005. The Team System Edition of VS 2005 provides a platform to build complex software suites and provides a vast number of reporting, monitoring and task assignments tools, to name a few, to all participants within a project. This greatly enhances the flow of information and intelligence when the team needs to take proper decisions. VS 2005 also provides the upper management multilevel overviews of the project to monitor and track progress. This report can be produced at will or at predefined times. The system can generate reports in multi format HTML, XML, Word, PDF or Excel and can post them on the web or email them to the interested
parties. The most important aspect of VS is that it is extensible and standardized. All the project and team information is kept in a database in XML format. This feature makes this information open for third parties to add new features to the VS 2005. For example, a third party can implement a new format for the reports generated. VS 2005 also provides an API so that organizations can integrate their compatible value-added tools into the VS 2005 environment.

4.4.2 Internet-based Services and Free Resources

DSL and Cable broadband networks are now within reach of regular consumers. Everyone living in developed metropolitan areas has technical access to megabits of network access bandwidth in their homes, offices and handheld devices. Digitization has moved from being a luxury to being a commodity. Internet-based service providers have recognized that trend and responded by opening their resources in various ways to empower and satisfy end-users (the service customers). This trend is of substantial impact and has brought along the convergence of technologies that created more services. This can be best noticed through the "proliferation of Competitive Local Exchange Carriers (CLEC), who can provide subscribers with a package of telephone and fax service, Internet service provision, cable TV programming and specialized data services."[1]

Online Banking and similar portals brought back-office services in front of the consumer. Similarly, many institutes and government services have setup web-based portals for the easy access and convenience of their customers. Many of these portals are standardized or built as a collection of services, which can be integrated with other systems. Of
prominence among those open systems is the Google Web Service which enables users to invoke searches on the Google search engine directly using web method calls and without having to access the website. The results of the search method are returned as a collection of objects of search results. Each object represents a single result entry or “hit”. The object has many member properties such as URL, Type and Domain etc. Yahoo!, The Weather Network, Reuters, Amazon and many others also offer similar free Web Services.

Web Services are proliferating as an industry standard for communication between web-based services. Tapping into these available Web Services adds value and cuts costs of software systems.

Also of great importance is that vendors of major software development tool, such as Microsoft, Sun and IBM offer the developer community many free resources, such as technical support, free complete unlimited use software for development purposes, and a growing library of whitepapers and webcast videos and technical demos. These resources empower software developers with two major abilities: the ability to “try or demo the product before buying it” and the ability to cut costs on basic set-up and mainstream training as most of the webcast videos and demos provide introductory level technical details.

4.4.3 Communication Infrastructure

The infrastructure focused on in the thesis hypothesis deals with mainly communication and data networks. These networks are disparate and include “telephone networks, data
networks, computer communication networks, cable television systems, and terrestrial and satellite broadcast systems” [1]. This infrastructure has been at this stage of robust advancement since well before the year 2000. The tremendous amount of bandwidth is something that needs to be exploited for the benefit of bandwidth-intensive broadband applications.

4.4.4 Standards

Standards play a major role in convergence of technologies and expansion of services. The most important contribution of standards that support service-based architectures is XML. XML answers two major concerns for systems running on heterogeneous platforms. By being pure text, which is supported by all platforms, XML answers the compatibility issue. XML, being strongly structured, is easy to parse and expand.

4.4.5 Best Practices

Currently Best Practices can be applied to the various stages of building complex software suites. Ranging from variable naming conventions to the best way to apply object pooling and recycling, normalization of database schema and other aspects are all now mature and available to the software designer community.

4.5 Broadband Multimedia Collaborative Spaces (BMCS)
Broadband multimedia spaces are systems that entail more than one dimension of data and require higher capacity access to the infrastructure. The discussion below will detail the two main features of BMCS.

### 4.5.1 Broadband

There are many definitions for broadband. They all converge at the idea of having multiple channels of signals over a single communication network connection. These concurrent channels provide a high-capacity, two-way link between an end user and access network suppliers capable of supporting full-motion, interactive video applications [93].

### 4.5.2 Multimedia

Multimedia is a term that denotes information that is presented using a combination of video, audio, images and text.

### 4.6 Collaborative Space (CS)

To some a CS is a web-based medium for the interaction and sharing of resources and knowledge. Comenius Multimedia Projects and Communication Technologies (COMP@CT), a European Educational Network, defines a collaborative space as “web-based applications and services that allow asynchronous communication and collaboration regardless of distance and, with some limitations, also time” [94].

Wikipedia is an example of such a definition as people collaborate on creating this web-based encyclopaedia asynchronously with no regard whatsoever for the participants place or time space.
Email, web forums, static web pages, Instant Messaging, Internet Relay Chat and Voice over IP (VOIP) communication tools such as Skype [95] are all common pieces of a collaborative space. No one would argue the power and benefits of using the Internet and web browsers for collaboration. A Web browser however takes away from the end-users a major set of sophisticated faculties and services catered by the OS residing within their computer systems. Browser-based applications, also called thin clients; have a limited wealth of capacities. On the other hand, desktop applications, also called rich clients have access to all the resources of the client machine. There are pros and cons for both types. In the year 2002 Microsoft proposed a new hybrid generation of applications, called Smart Clients that merge the strengths of both thin and rich clients. Thin, rich and smart client applications will be discussed in the next chapter.

4.6.1 Definition of Collaborative Space

\emph{A comprehensive collaborative space is a network-based virtual space that supports synchronous and asynchronous broadband multimedia collaboration. CS would offer guaranteed quality of service based on Service Level Agreements (SLA) furnished to the end-user by the supporting networks and service providers. The CS will have no regards to time and location difference between the participants. In other words, everyone will have the exact same experience and functionality and level of service.}
4.6.2 An Open-Ended Collaborative Space

The collaborative space will not be limited to the type of media files currently handled within a snapshot implementation, but rather gracefully and seamlessly accept and exchange all the various media files that are currently supported in the public domain.

4.6.3 Comprehensive Functionality

The collaborative space is to resemble a real life medium of collaboration except that it adds to it recording for the purpose of playback and it removes from it the restriction of time and location.

4.6.4 De-coupled and Flexible Design

The collaborative space follows best practices and object-oriented paradigms to ensure high rate code reuse and flexible design where changes can propagate globally within the code libraries. Abstraction and separation of concerns both horizontally and vertically also allow for the modularity of the components and allows for the future decoupling of the various tiers of the architecture.

4.6.5 Extensible Infrastructure

The collaborative space will be extensible both vertically and horizontally by means of adding or replacing feature and functionality, streamlining crosscutting concerns by means of employing an aspect-oriented approach. A software implementation of the collaborative space will allow its toolbars, menu bars, dialogue boxes, presentation layout frames, and all other components to be exchangeable and extensible without any
involvement on the part of the participants and all without losing previously implemented functionality.

4.6.6 Forward-Compatible Multimedia Type Handling

The collaborative space accommodates all media files currently available and should account for future development today. New media types emerging in the future will have to be handled in a way that is seamless to the end user for as long as an application that can handle the new media file is present on the end user machine.

4.6.7 Reusable Components

Many collaborative spaces are indigenous and propriety to the organization implementing or proposing them. A real collaborative space should in itself exhibit a high degree of collaboration. This can be achieved if the collaborative space is designed in pieces that are available to be integrated in full or in part into other collaborative spaces.

4.7 Service-Oriented Architecture (SOA) Based Framework

SOA is a new approach to solving old or recurrent problems of software reusability, scalability, security and openness for integration with other system. SOA is the best candidate that can handle many of the requirements mentioned so far in the building of a serious collaborative space.

SOA is a core component of the proposed framework and will be discussed in more detail in the following sections. Figure 4.15 below shows a basic SOA topology implemented

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
as a Web Service. All communication takes place following a service-request, service-response protocol using well-known standards such as HTTP and XML.

Figure 4.15: Basic SOA Model

The basic SOA model consists of: a service, a service provider and service consumer and a standardized communication protocol.

“A service is a software component that can be used in part of an overall business process. Services support a message-based communication interface, through which a conversation takes place. A service encapsulates its own state and business data, and it can be communicated with only through the service interfaces it exposes.”

“As the Internet and its related technologies grow, and organizations seek to integrate their systems across departmental and organizational boundaries, a services-based approach to building solutions has evolved.”[96]

---

According to a recent IDC study (June 2006), US “companies will significantly increase investments in service oriented architecture (SOA)-based services throughout the rest of 2006. The study, which analyzes the results of an IDC end-user survey of U.S.-based companies, reveals that the majority of respondents rated SOA as important to their overall IT and business strategy, and further indicated that they are planning to purchase SOA-related services by the end of 2006. IDC cites a significant opportunity for service vendors to assist companies in implementing SOA, and believes vendors must continue their education efforts, provide guidelines, and demonstrate business benefits to reach the majority of potential SOA users” [97].

4.7.1 Benefits of SOA

There are four major inherent benefits from using SOA.

SOA is a **Loosely Coupled** paradigm that allows the building of **Modular Systems** in a **Non-Intrusive** fashion and is based on proven **Standards** [98].

4.7.2 Loose Coupling

SOA allows for the quick assembly of process and functionality which simplifies debugging and integration and enables the reusability and sharing of process that had been developed at a previous stage which results in lower cost of production. Since services and applications consuming these services need not be residing on the same system, as is usually the case, different teams can be working on different parts of the
application and services on different platforms when necessary. In fact this can be done
ONLY if a high degree of de-coupling was maintained.

4.7.3 Modular Systems

Following a SOA approach for designing complex software systems leads to a
progressively staged and robust applications development. Consequently this enables the
avoidance of building complex systems that are made of large pieces of complex
software libraries. This allows for substantial reduction in complexity. Eventually,
building similar systems will be mostly the assembly of existing components and addition
of new features using newly developed components or services. The superset of
components in turn can be used as part of future releases or similar products.

4.7.4 Non-Intrusive

This is one of the key benefits of SOA. It allows the leveraging of existing infrastructure
and reduces and sometimes eliminates the need to re-write, re-factor, or re-test existing
applications. For example, consider a legacy database that is made of incompatible
schemas and that the contents of this database are needed by a new application or service.
Following the SOA approach, implementing a standardized service that hides the
implementation and details of the database and exposes services for access to the
database in request-response fashion would allow all other applications to leverage this
legacy system in a non-intrusive way. Non-intrusive means no changes will be introduced
to the database per se: a new layer of abstraction is built on top of the database to service its contents to the external interested parties.

4.7.5 Standards-Based

The power of SOA resides in that it is standards-based. Standards ensure platform independence. With platform independence integrating heterogeneous systems is possible with no regards as to which platforms the other pieces of the system run on. Standards are the common playing field or the common language that all applications can use to communicate with one another. Standards also allow for building scalable applications by leveraging and connecting to existing systems. And, by being predictable and consistent, they substantially reduce the complexity of building extensible and scalable systems.

4.8 Utilizing SOA for Building Extensible Systems

Service orientation solves the problem of scalability and security by means of optimizing access to the infrastructure. In a client server application clients access the server intermittently through opening two-way communication sessions. Many servers keep these sessions alive with the client until the client explicitly closes them or shuts down. Following this application design, supporting an extensively growing number of end users (such as Internet users) would require the deployment of many servers to serve those clients due to limitation in the number of live sessions that can be handled at once. One server could well handle however the grand total of all data exchanges between the client and the server. Figure 4.16 gives a simple client-server topology where the clients get responses to their query from the serving server. The server can be the actual database
server itself. In SOA the server would be the abstracted layer where the services are published and accessed.

![Image of a typical client/server interaction](image)

**Figure 4.16: A typical client/server interaction [99]**

### 4.8.1 Problem with Client-Server Approach

The Client-Server approach is not scalable to an Internet proportion and potential customer base. It is tightly-coupled, costly and imposes security problem.

### 4.8.2 Solution for 4.8.1

Following an SOA approach solves this problem by introducing a middle-tier service layer to act as a proxy between the client requests and server responses and would be the most appropriate solution as shown in Figure 4.17.

Following an SOA approach:

- Would allow thousands of users to log onto this proxy layer using their own accounts— for authentication and security— and submit their requests to the middle
The middle tier in turn would have a set of generic service accounts that would access the server and recycle these live sessions with the server to serve the growing number of end users. A Service account is an account, which handles the connections with the server and resides on the middle-tier or the data access tier. All clients through the middle tier will implicitly share this account.

![Figure 4.17: SOA Model as Replacement for Client-Server Model](image)

### 4.8.3 Problem with Single Services Middle-Tier

When deploying all services on one single middle tier, this tier becomes a single point of failure. If it becomes unavailable or overwhelmed by the requests then all users will suffer.
4.8.4 Solution to 4.8.3

This problem of a single services middle tier as described above can also be easily solved following the SOA approach. The middle-tier can split horizontally into a higher-level proxy and a lower service layer as in Figure 4.18. The proxy layer resides closest to the clients, dispatches requests to services deployed on the lower layer closest to the server and data. This splitting enables the higher tier to handle many more users, as all it will do is merely dispatch their requests. The lower proxies access the end resource or the server and do the raw data exchange, handle the request and process the data to the format that the end user requests, such as generating report of sales for a customer or final grades for a student in XML, HTML or PDF formats. The higher proxy dispatches user requests to the most appropriate lower tier proxy.

![Figure 4.18: Multi-Process Service Layers](image-url)
4.8.5 Problem with Multi Process Service

The process of finding the most appropriate lower tier proxy can be very complicated and CPU-intensive in that it would make the higher proxy very busy and occupied in calculating the factors and variables that would result in choosing the best server to provide the service. This would result in degrading response time and poor quality of service.

4.8.6 Solution to 4.8.5

The problem described above can also be solved following the SOA approach. The selection criteria of the most appropriate lower proxy to serve the user's request can be a service in itself. With that service up and running, the top proxy integrates with the new service that returns the address of the most appropriate service provider to respond to the user's requests. The top proxy can now do reporting services such as usage tracking, statistics on requests, reporting on quality of service, compiling feedback from service consumers while still able to handle more and more requests from end users. Figure 4.19 shows a split top-level service layer that makes a call to a helper service to decide which lower service is to be called to respond to the users' requests.
4.9 Rundown Scenario of an SOA Extensible System

Using SOA can solve many problems such as scalability and modularity. SOA is recursive. Which means the designers can decide to split a service into smaller components and services to the atomic level. This way the whole system can be primarily a superset assembly of the underlying services supporting the upper layer service consumers.

The following paragraph gives a rundown of the above scenario:

- Consumer A needs some service S.
- Consumer A contacts proxy Web Service P1 and places a request for address (URL) of the most appropriate middle tier proxy P2 to do business with.
- P1 returns the URL address of the most appropriate Web Service P2 to serve the Consumer A by either:
  
  A- Having its intelligent search engine perform the search criteria locally.

  OR

Figure 4.19: Multi-Tier Service Layer Model
B- Make a call to another Web Service that implements an intelligent search engine and requests the URL of the address of the most appropriate Web Service to respond to the requests of Consumer A. This part of the process is seamless to the Service Consumers.

- The Service Consumer gets what is the URL of the optimal Web Service that will best serve its needs.

Optimal services can be decided based on many factors, such as proximity of the Service Consumer, Service Consumer profile, security requirements, server availability, bandwidth, etc. The Service Consumer profile can be used to decide cheapest vs. shortest response time. Cheaper could mean minimal or no security (such as contacting the weather network) shortest could mean dedicated connection and therefore higher security at a higher premium. As described this "search engine" that would decide which is the best Web Service that suits the infrastructure utilization formula and the Service Consumer SLA can be more granular and far more detailed. Since a service is a function that does one and only one complete thing, this search engine qualifies to be its own service. Separating it completely from the higher proxy P1 would make the whole design more consistent with SOA. The Search engine in turn can be replicated to ensure that the system does not contain any single point of failure. Such robust and consistent SOA design would greatly achieve the two major objectives mentioned above.

Figure 4.20 gives a workflow view of the scenario above as the client contacts the proxy and the proxy then contacts the URL service provider. The service provider returns the
address of the most appropriate service to the proxy, which in turn relays it back to the client. The client now contacts the remote service using the returned URL following a basic service-request/response model.

Figure 4.20: SOA Usability Graph
4.10 SOA Final Notes

SOA with its standards and its inherent benefits can be the answer for future forward software systems. Integrating Web Services together creates what is called a mesh-up. This mesh-up offers a higher level of abstraction and is rich in functionality. SOA is geared towards integration of services into mesh-ups and this in itself is one manifestation of reusability and with standards replacing components, is guaranteed to be seamless.

Since change is the only constant behaviour of the field of Information Technology, a standardized approach to future application development and integration problems is the only viable and workable answer. Standards can no longer be accepted or expected to be a by-product of cycles of trial and errors but rather standards need to be the propeller of development and launching stage where the right process and tools are conceptualized and debated.

As more and more of software application behaviour, needs and challenges converge, Design Patterns can also be the answer for requirements of these applications. The dynamics of coupling standards and Design Patterns would for sure generate more flexible and robust process that can really stand up to serious economy-of-scale problems such as scalability, security and leveraging of corporate infrastructure.
4.11 Conclusion

The research statement identified the major characteristics of a meaningful multi-media collaborative space that can handle the demands of collaboration today and in the future.

It was found that the field of Information technology is divided into major fields of science that produce research results, which make more options open and more possibilities reachable. The hardware and software domains alike are at a stage where it is possible to build sophisticated software suites for broadband collaboration.

The statement indirectly identified the lack of such systems but proposed that a framework that satisfies the objective of the statement is still needed following a SOA approach. Adopting SOA is in line with the trends that are taking place now in the industry and hence add value to and trust in the proposed framework. There are certainly blind spots and risks in the process and they will be acknowledged and dealt with as they get discovered.

The proposed framework is discussed in detail in Chapter 5.
Chapter 5: The Proposed Framework

5.1 Introduction

This chapter deals with a more detailed breakdown of the thesis problem and provides details of the proposed framework to solve the thesis problem. It concludes with discussion of the overall proposed solution in a collaboration environment.

The thesis problem is first broken down into major logical branches and each branch is in turn broken into its subcomponents. This dichotomy allows for separation of concerns when dealing with the various part of a collaborative environment.

The proposed framework is a multi-tiered solution that leverages cutting-edge technology. The bottom-up approach offers easy conceptualization of the proposed framework. This also gives a visual reference on how the core and the various tiers (physical and logical) complement each other to meet the requirements and goals outlined by the research hypothesis in Chapters 1 and 4.

The identified requirements in the thesis hypothesis echo what has been identified in Liu, El-Saddik and Georganas (2005) [13]. Liu, El-Saddik and Georganas (2005) [13] specified six requirements that “must be given special attention in order for any generic framework to become a true dynamic model.” According to Saddik then, the framework
must support open architectures and interfaces, be highly interoperable for information exchange, be flexible, re-usable, and compatible with other systems.

5.2 Detailed Breakdown of the Thesis Solution

Since the thesis hypothesis deals with multimedia collaborative environments, it is necessary to decompose the problem into its subcomponents, or mini problems, and deal with them one-by-one. This will allow us to use a modular approach to devise the appropriate solution for the subcomponents. The decomposition will be recursive until a reasonable level of detail is achieved.

In multimedia environments, multimedia contents are exchanged and handled using appropriate tools. In the previous statement three major subcomponents or notions were identified, namely, the contents, the tools that handle and process the contents, and the delivery technology required for content to be exchanged.

Figure 5.21 below gives a pictorial breakdown of the thesis solution as follows:

- Starting with the thesis statement, we can identify that in multimedia collaboration, there are contents that are stored in some repository, tools to handle and process these contents and a delivery mechanism that performs and controls the content delivery process.

- The Contents branch identifies the contents management, maintenance and delivery processes. In this branch we tackle the challenge of managing different media types such as voice and video etc. Also the challenge of archiving, updating and keeping track of usage and access to those media files.
• The Tools branch shows tools that support collaboration. These can be front-end interactive tools or back-end tools that support the front-end tools.

• The Delivery mechanism mostly deals with the transport protocols that enables and covers all different modes of communication between the various external and internal components of the system.

Today’s technology and networks have matured to provide sophisticated tools, services, standards, best practices, methodologies and trends. These provisions constitute the context that can be utilized to build enterprise-scale collaboration environments (CE) that are standardized; forward-compatible. To achieve future-proof capability CE are composed from reusable and replaceable components.

![Diagram](image.png)

Figure 5.21: Solution- Sub Components

5.3 Multimedia Contents in Collaboration Environments
In digital multimedia environments different types of media content are used. These consist primarily of five specific media types, which are: video, audio, image, text and binary.

- Audio: sound files of various formats and compression schemes, such as MP3, Real Audio, and Wave.

- Video: moving pictures or frames (6-120 frames per second) with or without sound. To achieve the illusion of a moving image, the minimum frame rate is about ten frames per second [100].

- Image: still pictures of various types represented by compression schemes such as JPEG, GIF, etc.

- Text: the viewable content of a text file and anything that is made up of text. XML, PDF (Portable Document Format), HTML, and Microsoft Word documents are considered text. The text that the user sees when opening an HTML file for example, is the viewable file contents that are rendered by the effect of the tags embedded around the text and displayed when the file is “opened” by the appropriate reader that can understand the embedded tags. When the user sees something like **text in bold letters** that is due to the fact that the text string “text in bold letters” was embedded between <B></B> HTML tags like this <B> text in bold</B>.
• Binary files can be executable files such as .exe, java, DLL, COM files, or packaged and compressed files such as zip and RAR files.

The five types (audio, video, image, text, and binary) cover virtually all media types that get exchanged in a typical multimedia environment.
5.4 Thesis Problem Part I: Management, Maintenance and Delivery of Multimedia Contents

In a multimedia collaborative environment users are expected to receive and send multimedia files and contents. There is an inherent complexity in the maintenance and management of multimedia files due to various factors such as the files’ different purposes, target audience, size, formats, file types, and other characteristics. Most of the time, media files are managed by being logically organized and grouped in extensively structured file system directories as seen in Figure 5.22 below. This process, albeit intuitive and scalable, is certainly neither robust nor flexible enough and is complex to manage.

![Figure 5.22: Simple Directory Structure](image)
With potentially millions of files stored in a file system with complicated directory structures, the need for a simplified approach for system-wide maintenance and management and delivery of these files is extremely important. Sharing files in a formal setting requires an agreed-upon Service Level Agreement (SLA) between the participating parties. This makes it incumbent upon the contents owner, for example, to maintain an agreed-upon level of QoS (Quality of Service). Content owners then need to devise processes and procedures and put them in place in order to keep the sharing of the file contents consistent with the SLA terms. If the shareable files were kept on files servers this would lead to a complicated and slow process for the management and maintenance of documents and files.

For example, if the same directory structure is to be replicated as backup to recover from sudden file server shut downs or failures, there is no guarantee that the files in both structures, the original and the backup, are in sync or bit-wise equivalent. To access a specific file a comprehensive indexing mechanism needs to be in place, otherwise the user would have to perform a top-down search each time a file is needed. Figure 5.2 shows a typical directory tree structure of multimedia content files.

- Managing multimedia contents is becoming a progressively complicated process. The number and size of such files is growing at a very fast rate. It is easy to appreciate the situation by noticing the number of video and audio peer-to-peer file-sharing websites and tools such as Napster [101] and Bearshare [102] that are mushrooming on the Internet. It is important to say that managing media files is a
taxing business due the close attention required when handling different files with extensions. For example, keeping multiple exact copies of the same file consistent adds another level of complexity to the process. Further, keeping multiple versions of the same file also adds more complexity to the management process.

These are just two of the many challenges that plague employing a simplistic approach to managing multimedia files. This leads to the conclusion that the management of digital media files is not a trivial task.

Streamlining the process of managing and maintaining these files would simplify and therefore lower the overhead both in terms of cost of management and time to manage. Implementation of the transport for exchanging and sharing the contents of these files by using indigenous or common protocols is also simplified as a by-product of simplifying the process of managing and maintaining.

The challenge described thus far deals with:

1. Maintenance and management of the media files.
   - Management refers to tasks such as adding or modifying end-user profiles and access permissions, categorization and grouping of multimedia contents, collecting intelligence on usage and access, etc.
   - Maintenance deals with CRUD operations in database terminology.

CRUD stands for Creating, Retrieving, Updating (version control, bug
fixes, etc.) and/or Deletion (deprecating, obsolescencing, archiving etc…) of
contents.

2. Delivery of media files, which deals with ways to access and share their contents
   with end-users.

The most important aspect of multimedia files is the ability to share them. To increase the
rate of sharing for these files, a standardized simple interface is required that, once
implemented, assures that accessing and using these media files will be reasonably easy
to achieve.

5.5 Solving Part I of Thesis Problem: Streamline the Storage,
   Maintenance and Management of Multimedia Contents

This section deals with content branch of Figure 5.21. Solving this part of the thesis
problem requires:

1. The design of a detailed process to simplify and future-proof the storage, retrieval,
   and management of multimedia contents

2. Identification of the best-suited, scalable, delivery mechanism to share the
   contents.

In more specific terms, solving the thesis problem would require a generic methodology
framework that would streamline and simplify the development and management of
multimedia contents including software components that make up the collaborative
space. This would also require the framework to add an exceptional amount of control and intelligence to the infrastructure.

5.5.1 Solution for Handling Different Types of Multimedia Contents

To solve the problem of managing media files of different types is to make all multimedia contents unified and converge them into one single media type whether the media file content is image, audio, video, binary, or text. Also, the new unified single media type must be platform independent and such that it can be indigenously manipulated, consumed or processed without the requirement for any specialized tools. Platform independence makes the solution much less complex and more robust. So there is a need for a unified media type that is platform independent. In the following section more in-depth details about the unified media type will be presented.

5.5.2 The Unified Media Type

The single file type that is platform independent is pure text. And, the most suitable form of pure text is Base64 text as all of its characters are viewable on the screen and "since all platforms can decode and encode using this convention as 6 bit ASCII is widely supported. No special characters need to be dealt with" hence is best suited to be included inside a SOAP message. Base64 text is text made up of the upper- and lower-case Roman alphabet characters (A–Z, a–z), the numerals (0–9), and the "+" and "/"
symbols, with the "=" symbol as a special suffix code. The superset of possible values in Base64 is shown in Table 5.1.

Table 5.1: Full Character Set of Base64 Encoding

| ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/= |

Base64 text is a 64-character subset of the American Standard Code for Information Interchange (ASCII) character set. Pure text in general is the easiest form of data representation to process. Text is also completely platform independent. No matter what the original file type was before encoding (conversion), the file will be converted and saved as a Base64 text file.

5.5.3 Base64 Encoding

Base64 is a binary-to-text encoding scheme whereby sequences of binary strings are converted to a sequence of printable ASCII characters. The scheme is defined to encode a sequence of octets (bytes). The resultant Base64 encoded data exceeds the original in length by the ratio 4:3; the actual length of the encoded data is approximately 135.1% of the original [103]. After compression the Base64 text file size is virtually identical to that of the original file.

Table 5.2 below shows two examples of input from different media types and the output that results after the input is Base64 encoded. In the first row, a text input (taken from this paragraph) was encoded in Base64 text. The second row shows that the input was an
image file. The Full specifications for this Base64 system are contained in RFC 1421 [104] and RFC 2045 [105].
Table 5.2: Base64 Examples

<table>
<thead>
<tr>
<th>Original Input</th>
<th>Base64 Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>The single media type that is platform independent is pure text. 65 Characters</td>
<td>VGh1IHNpbmdsZSBtZWRpYSB0eXBlIHRoYXQgaXMgcGxhdGZvcm0gaW5kZXBlbmRlbmQgaXMgcHVySB0ZXhhLg== 87 Characters.... 65 *4/3 = 87</td>
</tr>
<tr>
<td><img src="https://example.com" alt="Carleton University Logo" /></td>
<td><img src="https://example.com" alt="Carleton University Logo" /></td>
</tr>
<tr>
<td><strong>File 5 KB before compression</strong></td>
<td><img src="https://example.com" alt="Some contents removed to save space as it spanned four pages" /></td>
</tr>
<tr>
<td><strong>File 5 KB after compression</strong></td>
<td><img src="https://example.com" alt="File 7 KB before compression" /></td>
</tr>
<tr>
<td></td>
<td><img src="https://example.com" alt="File 5 KB after compression" /></td>
</tr>
</tbody>
</table>

The multimedia files within this framework are any files that are considered tools or learning object content. Within the context of the proposed model for a shareable unified file type for all multimedia file types, every file would go through six main stages as follows:

1. The first stage: the file is created and saved on a local drive in its original format. This stage produced typical files such as a Microsoft Word document, a PowerPoint document, a Real Audio file, etc.

2. The second stage: the file is encoded into a Base64. In our proposed system each of the five file types will be converted into a Base64 text file.

3. The third stage: the file is added to the repository and a detailed profile is created for this file. This profile is metadata that captures all the information about the physical file and creates a summary of what the file contains. For example, it will tell what
time the file was created, what version it is, what type of media, and what type of subject it deals with.

4. The fourth stage: the end-user requests files from the repository.

5. The fifth stage: assuming that the consumer is entitled to access these media files, the file is fetched from the database repository then is encapsulated in a predefined package, and serialized over the network to the end-user.

6. The sixth stage: the end-user receives the file package. The package would minimally include the media file and its corresponding detailed metadata profile. The profile tells the end-user's application receiving the package how to process the file. The file is then decoded from Base64 to binary and then handled accordingly as defined by the included profile information. For example, if the profile indicates that the contents of the original media file are in MP3 format, the file would be decoded into binary and renamed into somefilename.mp3 and then the proper media file player would be launched. The consuming application would leverage the registry of the end user machine to know which application is associated with what file type in a process that is totally seamless to the end user.

Stages four and five both deal more with the delivery component of the solution.

Figure 5.23 below offers a flow chart overview of the process of converting the various media files into Base64 text as follows:

- When the tool receives a file it checks if it is encoded in Base64 and if not it starts processing it.
• The files get loaded and their basic information is captured such as name, type, extension, date created and the like. The type is decided based on the registered set of file extensions.

• The file is then opened in raw binary format and read as such, and then that binary stream is cut into words of 6-bits each and then converted into Base64.

• The Base64 files are then compressed and ready to be added to the database if the metadata profile associated with this file has been completed in full.
5.6 Benefits of Using Base64 Encoding:

Digital files are made up of binary values saved on digital storage devices. Opening these files in their binary format and encoding their binary content into Base64 text will convert all files into pure text. These Base64 text files get decoded on the receiver's side and back into their original format. It is then possible for any file to be exchanged, in its Base64 form, using the exact same mechanism that is used for pure text transfer. This
would also unify and simplify, i.e. streamline, all the processes carried out on these files such as compression, encryption, obfuscation and the like. With all media files Base64-encoded text, only one set of text-optimized compression algorithm, obfuscation algorithm, encryption protocol, and so on is needed.

This allows us to focus on a constant set of protocols that handle one single type of file and keep improving and optimizing it. For example, a compression protocol that works better for image files but not as well for audio files is no longer a concern in the proposed solution as there will be only one type of file to deal with. In this system there is only a need to improve the compression, encryption, and obfuscation of text alone.

5.7 Leveraging the Relational Database Management System (RDBMS) for Maintenance and Management of Media Contents

Now that all the files are pure text streams, the next step is to find the best data store in which to keep and maintain these files. Modern database systems are mature data stores that come with a full suite of tools that enable centralized enterprise data control and sophisticated indexing, backup, reporting, and data manipulation capabilities. In our proposed solution, a database is employed as the core data store for all multimedia contents, as well as for the software that can be part of the collaboration environment whenever possible.

The database acts as the data store repository, or the container, where these media files are kept in Base64 format. This format makes it possible to deliver the files in a
standardized, simplified way. The data store would also handle other services such as maintaining a log of which media files were delivered when and to whom; gathering real-time usage statistics; maintaining security; managing restricted access; fast failover recovery, and content indexing. These requirements are needed in order to make the management and maintenance of multimedia contents both robust and scalable. Since all the contents are Base64 text a Relational Database Management Systems (RDBMS) is the right choice to handle all the storage requirements and data gathering and processing services.

There are many gains realized by utilizing a RDBMS as the data store for media files.

• RDBMS makes it much simpler and easier to deploy, manage, and optimize enterprise data and contents and analytical applications.

• RDBMS provides centralized management utilities that enable data administrators to monitor, manage, and tune all of the databases and associated services across the enterprise.

• RDBMS allows for the management and optimization of enterprise-wide data from centralized consoles, thereby eliminating a substantial amount of work in upkeep of the enterprise data infrastructure.

• RDBMS can maintain metadata profiles for each of the files stored in it. This greatly enhances and expedites searching and indexing of contents in addition to being required for decoding the Base64 files back to their original format and extension. Table 5.3 below gives a schematic overview of a simple profile of various files saved in a database table.
### Table 5.3: Sample Profile of Base64 Object in the Database Table

<table>
<thead>
<tr>
<th>Object ID</th>
<th>Contents Description</th>
<th>Type</th>
<th>Object Name</th>
<th>Original File Extension</th>
<th>Base64 Encoded Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>GUI001</td>
<td>Virtual TA Room</td>
<td>GUI</td>
<td>Chartroom</td>
<td>DLL</td>
<td>AAAABEEEFFFR...</td>
</tr>
<tr>
<td>GUI002</td>
<td>Whiteboard</td>
<td>GUI</td>
<td>WBForm</td>
<td>DLL</td>
<td>CSEZAADDDDDDFR...</td>
</tr>
<tr>
<td>GUI003</td>
<td>Carleton standard logo image</td>
<td>Image</td>
<td>Carlogo</td>
<td>Jpg</td>
<td>R01GODPNAOYAP...</td>
</tr>
</tbody>
</table>

- RDBMS provide a high availability of data to end-users. An exact bit-wise copy of the database or database mirror must always be ready to take over the tasks of the failed database with seamless handover. This cannot be done using pure fileserver topology where backups for those files can be done, since backup and database mirroring differ substantially. File server back-ups are wholesale packaging of everything there is on those servers and this process lacks three major features that support a high level of data availability:
  - Files (DLLs, Media files etc) in the backup are not accessible individually, as they are packaged in Binary Large Objects or BLOB.
  - The backup process doesn’t recover from the failed file server without serious network administrator intervention. The process will be noticeable by end-users, as they will be affected by delays.
Backups are normally scheduled at non-prime time so as to reduce the interruption of service to an acceptable minimum. For example, backups are scheduled to take place at, say, 2:00 AM when presumably few or no activities are taking place on the file server. This 2:00 AM is time-zone-specific: a 2:00 am in Ottawa is 3:00 pm in Tokyo, a prime time where some users may want to access the file server remotely and do some heavy access. This attribute deprives any collaboration environment of one of its best qualities, which is being free of time-zone constraints. In other words, backups deprive the collaboration environment of the quality of being time-zone-independent and boundary-less. In fact, if users in Japan were required to adjust their schedule to an Ottawa-based environment it would as though they had moved virtually to Ottawa. That is what the framework totally avoids.

Database mirroring is a real-time process, while back ups are offline, and thus avoids the time-zone problem.

- Current database systems are better suited to handle environments that are not fault-tolerant. It’s no longer acceptable that performing scheduled database maintenance and sanity checks on data should require a system shutdown. Proper database mirroring coupled with automatic failover to a standby server would substantially increase the availability of the environment.
- Database systems now have built-in symmetric and asymmetric encryption capabilities that are robust and secure.
In summary, Multimedia Collaborative Environments (MCE) are inherently distributed and bandwidth and resource intensive. The Collaborative Environment is meant to be extensible and able to support thousands of users around the globe. That ability would require that the whole environment be replicated in more than one geographic region in order to maintain the same level of quality of service (QoS). Having thousands of users accessing the same physical resources creates bottlenecks in the model and consequently becomes a single point of failure, resulting in contention for bandwidth, which will consequently result in reduced quality of service and hence lower end-user satisfaction.

Having the Base64 files saved in a database and having a Learning Object Metadata (LOM) profile associated with each file would create a LOM library that can be part of a shareable Courseware Object Reference Model (SCORM). SCORM assumes the existence of services that have been launched and subsequently keep track of learner and learning object usage, what learning object to be delivered, and generate reports, etc. These data tracking activities can be made part of the database infrastructure handling the services. In addition, the actual learning objects themselves and the tools needed to work on those objects may be kept in the same database.

This database and the service layer on top of it constitute a comprehensive Learning Management System (LMS) that can be part of a global collaboration effort with standards and interoperability at its core.
Today's RDBMS technologies are the natural right choice for storing enterprise mission critical data where, in a collaboration context, the multimedia contents are the "data". It would be counter-intuitive not to use a full scale RDBMS as the core data store for the multimedia contents to support multimedia collaborative environments.
5.8 The Challenge of Delivering Multimedia Contents

So far encoding the binary stream contents into Base64 text has solved the different-media-type challenge. Also, employing an RDBMS solved and simplified the complicated process of management and up-keep of potentially millions of multimedia files. What is left for us to define is the best paradigm that would make this infrastructure open and accessible and interoperable with other systems. Hence, the challenge here is in selecting the most suitable mechanism for the delivery of the multimedia contents. Now that the learning objects and the LOM are in the database and ready to be shared there is a need to devise a way to make these learning objects shareable.

From this point onwards the database in the proposed model will hold the multimedia content and customized software libraries required for an advanced level of collaboration or system management and will also hold the metadata profiles and LOM of the contents all encoded into Base64 text. Opening access to these files to other external and internal parties so they could integrate them within their own processes and requirements would leverage the value of those files once integrated with the new systems.

The delivery mechanism sought must be consistent with the overall objective of the framework and meet all of the six requirements identified by the thesis problem. Most importantly, the delivery mechanism also needs to be platform independent, standardized, and modular to secure the infrastructure and hide implementation details from external parties.
The delivery of media files requires the deployment of technology that is suitable for those various types. Binary files, for example, can be downloaded using HTTP or FTP protocols. Image files can be retrieved and sent using the standard email protocol Multipurpose Internet Mail Extensions (MIME). E-Mail MIME may be used to transfer file attachments of any kind. However, the URL pointing at an image file, for example, may be pointing at an old version and the image file MUST be kept in the same physical location on the file system for, as long the URL is “public static”. This approach is very limiting and rather primitive. This approach also deprives content owners from leveraging the value of these files by offering more customized and personalized viewing and access capabilities.

These basic protocols themselves are not sufficient to provide dynamic and seamless access to the available files. Employing an increased set of protocols for different types of media files adds more complexity and defies the goal to simplify the process of delivering the multimedia contents.

Since the files are now in saved inside a database and all files are Base64 text format, this allows us to:

- Standardize, secure and optimize the way those files are accessed.
- Deploy of a small set of optimized processes to handle all the various aspects of handling and maintaining those files. For example, only one serialization interface for file transfer (text serialization) is needed, only one compression algorithm, one obfuscation algorithm, one encryption protocol, and so on.
- Serialize text only, which happens to be the easiest form of data to be serialized.
- Deploy Web Services for the delivery of the Base64 encoded text out-of-the-box, as text serialization is primitive to Web Services.

## 5.9 Employing A Service Oriented Architecture for Delivery of Multimedia Contents

A Service Oriented Architecture (SOA) is the best-suited delivery mechanism that meets all the requirements and qualities highlighted in the previous section. Since SOA requires implementation of a standard-compliant serialization interface for each of these media types exchanged through the service, having the media files encoded into one single type greatly simplifies the serialization implantation.

A Service-Based Architecture (SOA) is an architecture that is made up of a Service, a Provider of the service and a Consumer(s) of the service. An SOA is a departure from the Object Oriented Paradigm (OOP) where data and functions that operate on that data are encapsulated together into classes of objects. OOP presents a higher level of coupling than of SOA.

In SOA, data and functionalities are decoupled. Data becomes a resource (a product) that gets exchanged between the Service Provider and the Service Consumer. This exchange takes place in a standardized way. The participants know how to communicate, request service and receive the service. That requires that the data sent and received over the network be encapsulated in a predefined packaging that both players can understand. It is natural to see that standards are the heart of such architecture.
SOA follows the Service-Provider and Service Consumer topology. Figure 5.24 below is an example of a SOA topology where the service provider has published the service definition. The service consumers subscribe to the service and download the service interface, which defines the way to invoke the service. When the consumer needs the service, it issues the service request message (SOAP message). The provider invokes the service locally and channels the results back to the consumer.

![Diagram of Basic Service-Oriented Architecture]

**Figure 5.24: Basic Service-Oriented Architecture**

Web Service is a current implementation of SOA. In Web Service topology, a Word document cannot be sent in its native format over a Web Service message unless the Word document object has a serialization interface implemented within the Web Service itself or inherited from an external library that implements the serialization library of the Word document object. The exact same scenarios would apply to most other media types, such as PDF files, Executables and so on.
Considering that new multimedia types may become available, the Web Service will have
to be expanded to accept these new media types and hence integrate the new media XML
serialization. There is no way to have the same service handle the exchange of files of
different types using the exact same serialization interface. However, after encoding the
new media type into Base64 text, the same process that handled the currently known
media types would handle the new types with no modification of the software tools, in
which case, just the proper profile for the new media type is required to be identified in
the database.

A true generic model should be future-proof and forward compatible without limit and
without the need to ever again touch the code implementation of the Web Service.
When all contents are in pure text, SOA deployment becomes the easiest choice as XML
serialization of types other than text is necessary to have the exact same service deliver
all types of media types including the new ones the would come in the future. This will
produce a future-compatible and future proof standardized solution for the delivery of the
multimedia content.

The sections below present the proposed SOA model for the delivery of the media
contents.

5.9.1 The Service Provider

The service provider would be a software agent interface: a Web Service that would
present consumers with the lists of services and files that can be serviced. Note that the
Web Service is acting as a proxy or an agent that makes accessing these media files possible to the entire potential customer community.

### 5.9.2 The Service Consumer

The service consumer is the major beneficiary of the whole system and perhaps the most important player as the system objective is to make all the resources available for this customer to use. The value of a service is determined by the number of consumers that consume the service and the frequency these consumer require the service.

### 5.9.3 SOA Data Access Layer

The data access layer is a service-oriented layer where all communications and interactions between the database infrastructure and the end-users are handled in a standardized way. This makes the solution scalable and secure. Interaction with this layer is purely through service-request calls. No direct access to the database is allowed. This would make the solution secure because the database will be hidden from direct access by end-users. It also makes the solution scaleable and, as it could be created as a single point for the management and capturing of requests, could therefore be used to increase security or introduce improvements. On the other hand, this single point of entry also creates a single point of fault. This problem can be overcome by deploying a service oriented solution whereby the proxy service interfacing with the request would interface between the end users and this layer and dispatch requests to the available database.
Figure 5.25 shows the proposed model to solve the first part of the thesis problem, which can be described as follows:

The potential user community of an online collaborative environment is global. These users would have access to the service provided by the collaboration infrastructure through service-request packaged in SOAP messages since the external service layer is pure SOA based model. The service layer accesses a lower supporting internal service layer again following pure service-request and service-response protocol. The internal service accesses the data repository using database-specific calls. This ensures the system is purely service oriented and changes can be introduced in a seamless fashion in regards to the end-users. This also protects the database from direct access by the users and external service.

The solution shown in Figure 5.25 assumes that there are multiple databases, which are identical and mirrored to maintain bit-wise equivalency between database contents. Some of the services that are available at the External Services Repository Layer are:

- Contents profile service, for the creation and management of media content profiles.
- Database contents browsing services.
- Record management services, to add, remove and update records in the database.
- User profile services, for the creation and management for all types of users.
- Application profile services, for the creation and management of the profiles for all types of applications accessing the data store.
- Statistics and reporting services.
Figure 5.25: Part I Proposed Model

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
5.10 Thesis Problem Part II: A Generic Tool for Multimedia

Collaboration

This section deals with the second branch of the thesis problem: namely the collaboration tools. Multimedia collaboration requires the presence of tools that handle the various types of data that get exchanged between the various participants, as well as tools for multi-way collaboration such as chat terminals. Most collaboration tools investigated are built from the ground up as a whole unit and not decomposed into separate components that can be integrated into other systems. This approach of building collaboration tools deprives collaboration of its best quality; which is "to share resources". Resources in a collaboration context are not limited to content exchanges but also include the tools used to exchange contents.

There is a clear need for a comprehensive collaboration environment that is flexible and interoperable with other systems and consequently can have access to external infrastructures and, in turn, provide open access to its own underlying internal infrastructure.

This cannot be achieved without the presence of a generic framework that identifies the various layers of communication; the various tools that are needed; and the various modes of interaction between the participating parties in the collaboration environment.

To solve this requirement, a generic framework will have to be identified including its various pieces in order to make the system and its components flexible and expandable.
It is also of importance to identify what current technologies can enable such a system to be implemented whenever possible without having to produce or request new tools.

5.11 Solving Thesis Problem Part II: The Generic Framework for Collaboration

We will start by identifying the various modes collaboration takes place in. There are two main modes, namely real-time or synchronous and offline or asynchronous. Some of the underlying network services certainly support both modes, such as multicasting, bandwidth management, safe failover and recovery services, authentication, and user access restriction to name a few. There are also mode specific services such as IP roaming and discovery where a mobile user using dynamic IP addresses can be found for a real-time multi-way chat session, for example.

The more we think in abstraction about the framework for collaboration the more services are identified and since the future can introduce new modes or new facilities that are to be integrated into a comprehensive collaborative environment it becomes more incumbent upon us than ever to think in broad terms and not in isolation. Integrating mature and well-established tools and services into the framework and avoiding to “reinventing of the wheel” is mandatory and not optional. The framework hence should lead to the creation of a realistic and comprehensive superset of standalone and well-defined functional subcomponents. Also, the framework should allow for the possibility of expansion vertically through the addition of virtual layers of abstraction, or horizontally in terms of adding more services at the same level of abstraction.
In the solution described in Part I, SOA was used to capitalize on its many inherent benefits. The same benefits apply here in solving Part II. SOA applies too since this part deals with pure software application environment. These environments are suited to be built as groups interacting components as shown in Figure 5.26 below. Following this paradigm, the final collaborative environment will be a collection of small subcomponents that are “glued” together by standardized interface technology.
Going back to Chapter 3, a service is a standalone and complete function that is independent and context free of other functions. This leads to the defining factor of the solution, which is to build a superset of services resident in a service layer or service repository. This repository will have to be accessible both for browsing to discover what
services are available and also for integrating with other systems in an interoperable scalable fashion that hides the details and maintains flexibility and a high level of availability.

The four main technologies used in this framework are:

1. An N-Tier stacked model, described below.
2. RDBMS as its core data store.
3. SOA as the standards for communication and interaction between the various distributed components.
4. Smart Client technology for the front portals.

In the following sections, the details, extent and limitation of each of these technologies as part of the overall framework will be presented

5.11.1 N-Tier Stacked Model

The framework is a stack of abstracted layers and therefore the framework can be described as an N-tier model. Although some argue that N-Tier indicates physical separation [106], in our model N-Tier indicates the logical separation of logically grouped process, functionality and domains. N-Tier models offer inherent benefits of separation of concern, which plays an instrumental role in simplifying the complexity of the proposed framework. In addition to allowing for separation of concerns, adopting an N-tier model offers many visible and practical benefits as it:

- Permits development both in stages and in parallel.
• Permits the swapping of components in seamless fashion.
• Helps in identifying bottlenecks.
• Simplifies the conceptualization of the system.

These and other benefits made the selection of an N-tier model the right choice for the proposed framework.

5.11.2 SOA between Tiers

To keep the model standardized and to increase the degree of flexibility, Web Services technologies are deployed as the main mode of interaction between the various layers and components both horizontally and vertically.

Located on top of the data store layer are layers of abstraction that act as the service layer of the model. The service broker is the main interface with this layer. The broker accesses the layer service layer repository and dispatches services to the service consumers by returning the proper URL location for the service as well as multiplexing requests and messages.

The service layer is conceptual, which means it can be multi-layer in itself. SOA technology was discussed in detail in Chapter 3. SOA offers standard processes to communicate between distributed systems and components in a platform independent fashion. SOA enables the framework to be open and receptive for future expansion and change. In fact a pure SOA approach makes the system boundary-less.
5.11.3 Smart Client Presentation Layer

The top layer is an open-ended collection of Smart Client applications that are used for collaboration and to interact and communicate with the lower layers for various purposes depending on the users’ roles, context and domains. Smart Client is a new cutting edge technology proposed by Microsoft as an answer for global reach and security of applications.

It was “built to take advantage of new XML technologies. Smart Client software delivers the best of both worlds, combining the reach of the Internet with the power of local computing hardware.”[107] XML is at the heart of Smart Clients and Web Services (an implementation of SOA) are considered the foundation of Smart Clients. There are other technologies and models used to implement the presentation layers.

The presentation layer can be designed as one of three types:

1. Thin clients, with the business logic and data access layer resides physically separate from the thin application. Most thin client applications run within the context of a web browser. There are three main benefits of using thin client applications:
   a. Ostensibly this provide a secure environment – as secure as the web browser is in fact—and hence the application is limited to the resources that the web browser is allowed to use. Generally speaking a thin client can only access the temp directory and any subdirectory below that. Thin clients cannot access anything outside the temp directory and also cannot access or make changes to the file system or the operating system registry.
Thin clients cannot invoke anything outside the web browser without the physical consent—such as clicking OK—of the end user. Therefore, thin clients lack the ability to capitalize on the resources available on the running machine, such as a common dialogue box, access to hardware and RAM optimizers etc.

b. It is also very important to mention that pointing the web browser to the URL where these applications are deployed runs such applications. An example of such application is Banking Online portals.

c. A third important benefit is that since thin client applications are web-based, all users use the same application version that is publicly available by the application URL. There is no need to download patches or upgrades, as there is no application installed formally on the user’s computer. Every time the user visits the URL, the user is accessing the latest version of the application.

2. Thick clients, also known as rich clients, have access to all the resources of the system where they are executing. This provides richness in the look and feel as well as in functionality. Thick clients’ ability to access whatever facilities are deployed on the system offer a higher level of flexibility at design time. These clients, as the name suggests, are heavier than the thin clients and also less secure. Normally the business layer and data access layer reside on the user machine when a thick client is being used. Since thick clients have full access to the file system and system registry they pose a bigger security risk than thin clients.
Normally thick clients are machine dependent—except for java, if the proper libraries are installed—and require more attention and frequent updates that need to be administered and scheduled.

3. Smart Clients, a recently released (2002) technology developed by Microsoft, are a mixture of the thin and thick clients. Smart Clients offer the best features of the two models. They offer the benefit of having an updated application each time the user accesses the deployed application. Smart Clients run within the context of a web browser security model and hence are as secure as a web application. In addition they are similar to Thick Clients when it comes to accessing the file system and other resources available on the running machine. Smart Clients tend to have a smaller hardware footprint than thick clients.

After investigating the three main models for the presentation layer, Smart Client is the cutting-edge technology that was adopted.

5.11.4 The Service Layer

This is the most important layer of the whole model. It is here that all the different components that can be integrated into other systems in a standard way can be deployed in real-time. Figure 5.26 gives an overview of the overall framework for comprehensive dynamic collaborative environment as follows:

- Layers offer a level of abstraction that allows the introduction of new services and making them available to external parties in a seamless fashion.
• It is a pure service-request, service-response model. All the communication takes place following a well-established standard.

• The service gateway acts as a one-stop shop for integration. It catalogues the various tools and services available for sharing. For example, the web search can be catalogued for integration as either a standalone tool or as a Web Service.

• The end-user can decide to actually get the full tool downloaded and integrated into their system, or get just the URL address of where the Web Service resides.

• If the user wanted the actual tool downloaded then the gateway offers access to the underlying service that would handle the processing of the end-user request.

• If the end-user wants to just integrate remotely with web search Web Service, the gateway would return the URL address to where the service is deployed.

• The end-user will not know if the tools are downloaded from a server next door or from a server across the globe. The gateway offers a complete layer of abstraction.

• Below the gateway is an abstraction that would hide the actual tools and service absolute source, be it an external or internal service or tool. This topology can occur when a consortium of institutions can join hands to create a mesh-up of services to leverage their multimedia content value and infrastructure. This layer would interface with their corresponding service and make them available to the gateway residing on top of it.

Following this approach can be recursive and hence can be expanded following a simple paradigm of abstraction and details and implementation hiding. This would
enable the framework to expand or shrink for that matter without requiring the end users to change the way they communicate with the gateway layer. Deploying Web Services ensures that all requirements are met as Web Service support sharing or resources as well as interoperability and platform independence.

5.12 Conclusion

The solution to the thesis problem utilized four main technologies, namely, RDBMS as the core data store for both the software and multimedia consents, Base64 encoding to unify all multimedia types into Base64 text, Web Service as an implementation of SOA, and Smart Clients at the presentation layer for the end-user to integrate and take advantage of the SOA used.
Chapter 6: Collaboration in a TeleLearning Environment

This chapter is divided into two sections. The first section deals with the conceptual model of a collaborative space in the context of a TeleLearning environment including its concepts and main subcomponents. The second section presents the prototype implementation of subcomponents based on the framework proposed in Chapter 5.

6.1 Introduction

It is rather impossible to implement the full context of collaboration due to its many forms, facets and domains. Some forms of collaboration, however, are more obvious and intuitive than others. Certainly the classroom environment is a space where the collaboration factor is very high. By collaboration factor we mean the amount of collaboration activity relative to other activities. For that matter, a TeleLearning setting is chosen as the context for developing proof-of-concept prototypes based on our proposed framework.

TeleLearning is the art of delivering and acquiring knowledge where distance, time and the physical space are no longer detrimental factors to the quality of education.

There are two core concepts within a TeleLearning environment:

- The medium where knowledge is exchanged. This we refer to as The Virtual Classroom Environment or (VCE)
• The content that get exchanged within the VCE.

6.2 Definition of the Virtual Classroom Environment

When talking about the virtual world it is necessary to define a reference to the real world so that the concept and ideas will be easier to understand. In the field of education, the classroom has been the most important space for collaboration. Teacher and learner activities predominantly take place in classrooms. Teachers make use of tools available for them to deliver the subject matter to the students. This is the real classroom; the virtual classroom mirrors these settings, adds to them and sometime subtracts from them.

The very essence of building a virtual world is to take the traditional methods, tools, processes, business rules and concepts from the real world and transform them – by means of information technology and digitization - into their equivalent virtual counterparts. This transformation in many cases enhances the quality of the original concept being transformed. In a Virtual Classroom Environment (VCE) setting, this enhancement will result in improving the quality of teaching a learner would receive.

One last point still: the term TeleLearning implicitly reflects the learner side or the student side. However the full picture and concept addressed here is a TeleLearning-TeleTeaching Environment where knowledge is delivered by a “TeleTeacher” and is acquired by a “Tele-Learner”. Before going further into the actual component specification and design of the virtual world, it is very important to give a concise definition of the virtual world in the context of the TeleLearning-TeleTeaching field.
The online distance-learning (TeleLearning) environment is a virtual collaborative space that utilizes technology to emulate and improve the experience of the learner in real-world environments.

6.3 Scope of Online the TeleLearning Environment Definition

In this section we will identify the scope of the various concepts and terms that appeared in the definition above.

**Technology:** Information technology, such as network technology, data storage, search and retrieval technology, software and hardware technology, etc.

**Emulation:** The virtual world should at least match the quality of education acquired in the real world. Round table discussions, teacher's feedback, Question and Answer sessions, teaching assistant office hours, all must be of equal or better quality than that of the real world. Virtual tools such as virtual blackboards, audio-video conferencing between teachers and other learners, etc., must also be of equal or better quality than that of the real world.

**Improving:** This is the most important quality of the virtual reality. In a perfect world, a comprehensively implemented online learning environment utilizes a reliable multimedia broadband network with adequate bandwidth so that distance and time zones will no longer impede the proliferation of knowledge. Students will enjoy the playback feature of multimedia lectures. People with physical challenges for whatever reason will have an equal opportunity to access knowledge and information like everyone else. The word
comprehensive is important as the more attention we pay to the details in implementation the more user acceptance we reap.

Lytras and Pouloudi [108] pointed out that highest rates of distance course dropouts are due to insufficient effort in realizing the full potential and value of E-Learning. They also argued that the available technologies expand from time to time the alternatives and choice of implementation and conceptualization of the software systems, but the core competencies are constructed around clear ideas and underlying concepts. This paragraph is not meant to cover all the benefits of online education and collaboration. In fact, one can write a book on the benefit of well-thought-out on-line TeleLearning spaces.

Table 6.4 below provides a reference to the main user experience in the real classroom and the corresponding virtual-equivalent. Table 6.4 also highlights the enhancements that the virtual experience affords the learner and teacher.

**Table 6.4: Real and Virtual World Experience Comparison**

<table>
<thead>
<tr>
<th>Real World Experience</th>
<th>Virtual World Experience</th>
<th>Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Face-to-face experience such as class attendance and round table discussions, one-on-one meeting.</td>
<td>Multimedia and Video-Audio conferencing</td>
<td>Session recording. Playback allows absentees to stay at par with the other participants.</td>
</tr>
</tbody>
</table>

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
<table>
<thead>
<tr>
<th>Real World Experience</th>
<th>Virtual World Experience</th>
<th>Enhancement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Course handouts, notes, assignments, test results, test solutions, resource, and bulletin board.</td>
<td>Web page, E-mail, Downloadable documents</td>
<td>Accessible globally, update-to-date information, available instantly</td>
</tr>
<tr>
<td>Black/white board</td>
<td>Virtual blackboard</td>
<td>Session can be recorded and printed later as part of course notes</td>
</tr>
<tr>
<td>Office Hours, Q&amp;A sessions</td>
<td>Internet Chat Relay (IRC), Instant Messaging (IM) or news group can be designed in a way that emulate office hours experience</td>
<td>Session recording. Ability answer, comment and share answers with many students at once.</td>
</tr>
<tr>
<td>Assignment hand ins, test writing</td>
<td>Softcopy submission via e-mail. Web-based tests</td>
<td>Deadline can be easily enforced. Test results available instantly.</td>
</tr>
</tbody>
</table>

Converting an educational institute into a virtual institute is a very complex process that entails the utilization of all software best practices, designs and techniques in the field of Information technology. Educational institutes are sophisticated organizations with business rules and processes governing every aspect of their daily business. Digitization or automation of these institutes requires far more than just software programs. The automation process in many cases requires the re-engineering of the actual business rules and processes to fit the overall model of the institute. Often time this incurs more detailed...
designs and redesign but most of the time this re-engineering step uncovers redundancies and inefficiencies with the education institute.

Our focus in this thesis will be only be on the education cycle side of the institute and more particularly on the tools that are needed to deliver knowledge from distance using digital medium.

But the TeleLearning environment also goes beyond being an emulator of the learning experience of the real world to rather be a sophisticated system of services that support the knowledge exchange process in full. The virtual environment, for example, needs to address the issue of students’ evaluation by an authorized evaluation body from the educational institute. The evaluation process ensures that the students are qualified to register in a specific program and hence be able to sign up for a specific course.

This evaluation process, which seems rather easy to implement in the real world, in practice is a complex process in the virtual one.

By now it is obvious that the definition of the virtual learning environment in Section 6.2 is limited to the part dealing with the exchange of knowledge is an E-learning environment. The definition constitutes the educational component of the online learning environment where the actual knowledge exchange takes place. Educational component can be further broken down into other subcomponents. Our discussion avoids the pure
The registrarial and administrative sides of business in the TeleLearning environment, which albeit important, fall outside the scope of the thesis problem.

The following section identifies the major components and players that makeup an E-learning environment.

### 6.4 Components and Players of TeleLearning Environment

There is a set of components and that make up the complete ecosystem of the TeleLearning environment. Figure 6.27 provides the tree hierarchy of the system users.

![Figure 6.27: E-Learning System Users](image-url)
6.4.1 The Learner

Any person who uses the TeleLearning environment to receive knowledge is considered a “learner”. Learners are the main beneficiary customers of the TeleLearning environment, since time and distance no longer impede their access to quality knowledge. In addition, since knowledge and information are exchanged in digital format, the learners have more control over how to view (zoom-in zoom-out, font size, screen size, colour schemes, etc...) and process (subtitle, translation voice-only, etc...) the information they receive. That is why a comprehensively implemented TeleLearning space needs to address the various scenarios and the various needs of the learners.

6.4.2 The Course Conductor

The teacher or the teaching assistant is that dispenser of knowledge and the conductor of the various pieces of information that a learner should receive. Teachers’ needs require special attention when the TeleLearning space is being designed. Features like office-hours scheduler, Teaching Assistant duties, assignments, tests, correction and marking all fall in the core business of the teacher and they need to be addressed when it comes to automating the process of knowledge dispensing.

6.4.3 The Content Author

This can be the teachers themselves or any designated person with the proper privileges. They design the course, create style sheets and templates and ensure uniformity in the look-and-feel across the various courses.
6.4.4 Administrators

Individuals who perform back office housekeeping work such as records keeping, finance, final grades, course scheduling and the like. These activities are not part of the TeleLearning environment per se, but are part of the bigger picture of distance learning environment.

6.4.5 The Course

This is the crown jewel of the TeleLearning environment. The value and quality of a course attracts learners from all over the world. The course value is the main differentiator between one institute and the other. And by extension, the quality of courses is indicative of the quality of the degree an institute offers. Hence, real effort must be invested in molding the courses in ways that make these courses appeal to the students. The course must be presented as useful, useable and sharable in order to foster collaboration in a macro scale.

6.4.6 The Tools and Services

The tools and services represent the bonding mechanism or the bridges between the learner, the teacher, the course and the administration services. The Tools and Services are the medium where the initial and subsequent contacts occur and where all the collaboration and knowledge exchange take place. Many invisible but essential services support the TeleLearning environment such as regular archiving of records and student account updates, status collection and performance monitoring.
Evidently in the virtual reality tools and the underlying infrastructure and supporting services are essential for the realization of the benefits associate with TeleLearning. Simply put, without proper tools and services in place the learner, teacher and course cannot connect.

Comparing a TeleLearning space to a real world example, the tools and services would be the foundation and structure of the home, the courses are the furniture and teacher and learner are the residents.

6.5 Designing the TeleLearning Environment

Best practice design and implementation techniques must be used during design process but that will not necessarily yield a useable virtual environment. There is a big difference between how software designers think and how teachers and students think. Teachers think in real multi-dimensional detailed objects (length, width, height, shape, colour, mass, weight, time, location, etc...) whereas software designers think of abstract concepts (Tier, Layer, Template, Master-Details Relationship, Is-a, Has-a, Templates, Interfaces, GUI, Data Stores, Service, Tool, protocols, algorithms, etc...). Marrying the two ways of thinking of the teacher and the software designer is critical in order for the designer to capture the real requirements and in order for the teacher to relay the real requirements.

For example, when a teacher is talking about what he or she teaches, they talk about all the various courses they teach as individual and independent units. They do that so that the listener understands the scope of the amount of work and effort they put into teaching.
On the other hand, if the listener were a software designer, he or she will be thinking that all those courses are nothing but multiple instances of the object "Course". The designer then tries to find commonalities in those courses in order to define a mega blueprint (the class) that describes the course.

Following the previous logic leads to the preliminary conclusion that the Object Oriented paradigm should be the process of abstracting and designing the TeleLearning space.

Students, courses and teachers are all distinct objects. Course registration is a component that does a specific task, namely registering a student in a course regardless of which discipline a student belongs to. Such crosscutting concerns can be better handled by applying Abstract-Oriented Programming, which was discussed in Chapter 3. Course registration relating components and similar components must be designed as a stand-alone, sharable and interoperable (using well-defined and standardized protocols) with other components to ensure reusability of code and functionality and services.

It is interesting to note that the whole online learning environment can be viewed to have different focal points. These different vantage points affect the design and functionality of any system. In other words, one individual can view and design the system with the course as the focal point. Another can design the system with the student as the focal point.

We argue that the strongest viewpoint should focus on the learning process and its associated players and service as one system. A TeleLearning environment designed to train fighter jet pilots is much different than the one designed for pre-schoolers.
Regardless of the different viewpoints, a thorough definition and sound design will most likely yield the same set of functionality and services being offered by both systems. Simply speaking the core infrastructure and designs of the systems will be equivalent.

The focus of the thesis problem views the “course” as the focal point of the design as shown in Figure 6.28 below.

![Course-Focused TeleLearning System Design](image)

Figure 6.28: Course-Focused TeleLearning System Design

6.6 High Level Description of the Prototype
In Chapter Five we defined the course contents as learning objects that are associated with a complex LOM and SCORM Metadata schema and stored in an RDBMS data store in a unified Base64 text. For that matter the first tool to be developed will be the tool that supports the creation and maintaining of course contents, which are:

- Content and User Profile Manager (CUPM): used to generate a metadata profile for each course and each tool stored in the learning database.
- Base64-Encoder-Decoder (BED64): handles the encoding and, optionally, the decoding of any file contents.
- Content CRUD Gateway (CCG): performs the basic Create, Retrieve, Update and Delete (CRUD) of contents. This tool is used only by authors of course content and collaboration tools and not by students. Enables us to add contents to the database and make them available for sharing.
- Course Browser: This tool allows the user to browse through and filter the courses that are now available within the database.

Following that we will move to the implementation collaboration tool. That tool enables plug and play addition and removal of services and tools and the switch on and off of services, which are:

- Chat Window
- Web Search Tool
- Virtual Classroom Space (VCS)
The Main Collaboration Shell (MCS). **This is the main proof of concept.** This container acts as the platform where all the reusable components can be integrated.

The various components are implemented and deployed as proof of concept and all tools are SOA-based that use pure service request and service response calls for communication and data exchange.

In the sections below the technical specification and design will be presented for each of the tools named above. Screenshots of the tools GUIs are shown also.

### 6.6.1 The Learning Contents, Applications, and Users Profile Manager

This tool is essential for anybody who will be involved in collaboration within the proposed framework. Users, contents and applications must all have been profiled in order to be able to access and use the infrastructure. There are some tools that do not require a user profile such as when the user is just browsing for courses or using the customized web search tool for example. These two tools are made available for ordinary users with no user-profile required. However, if the user is trying to register in a course or if the content owner is trying to add content to the database, a profile must be created for that user before such actions can take place. The user profile identifies what privileges a user can have when accessing the system. Student profiles could be auto-generated by the educational institute.
Every application and tool used within this framework must also have an application profile. Contents also require a content profile before getting saved into the database. Hence, we have so far identified three types of profiles, namely, contents, users and application. Figure 6.29 shows the breakdown of various profiles within the system. This profile manager tool is really the main starting point for any serious participation or usage of the collaboration environment. Collaboration needs contents and tools and users and those contents, tools and users all require profiles.
For example, a content owner uses the profiler tool to create a profile for content.

Assume that the content is an image file to be shared by the end-users. We can assume that the content author has had his profile created in the system, and then the profile would allow them to logon to the system with the privileges assigned to the user. The profile, implicitly affects what tools and services and menu options can be made available for content authors. Therefore, the profiler tool allows the content owner and author to create a profile for the course contents item by item or as a group.
For example, if there is an image file that is part of the course, the content owner would have to create a profile for that file so that it can be uniquely identified, tracked, retrieved and shared. Once a course profile is created, the author can assign contents to the course. The profiles are kept in a database table as descriptive metadata. The content profile metadata are file name, type, extension, version, owner, author, date created, and date last updated. There is no limit to what can be kept in terms of metadata about any content. However, this information is sufficient when the file is retrieved and displayed by the end-user. Once the profile is created, a unique number is generated and assigned automatically to the profile. Using this unique number the content owner can easily update a profile. Also this unique profile-ID helps in indexing the content of the table to improve access and retrieval time.

Figures 6.30 and 6.31 below show the main GUI of the profiler of a user and course respectively. The same tool can be used to create profiles for, courses, course conductors, administrators, or contents. The top menu dropdowns allow the tool user to choose which type of profile to be created if the use has the proper authority.

An application profile is an ingenious and creative way for controlling application functionality from a remote location. When a user launches the TeleLearning software application within the framework, the first thing the application does is to download its updated profile from the remote server service through a service call to the backend application profile manager Web Service. The application profile can be serialized in real-text or a DLL, or INI file.
Figure 6.30: General Profile Manager-User Profile Entry
An initialization file, or INI file, is a configuration file that contains configuration data for Microsoft Windows based applications [109].

The application profile holds information that specifically tells the application what menu options to display, what components are available for that user and the latest version of these components. If, for example, some service such as Profiles Manager is not available or inoperative, the application profile that has been downloaded by the application on a user’s logon will make this service appear disabled or hide the menu option that triggers it. If an option were available in the menu bar, then the service that gets triggered or invoked by that menu option is functional - otherwise the option would not be available.
Availability can be decided by various factors, such as user privileges or whether the service itself is up and running or not.

A user profile identifies the privileges a user can have when dealing with the collaboration framework. A user can be a content author or content user or course conductor, TA, Administrator, etc. The content author can be course content or software tools author. On the other hand, a contents user can be a teacher or the learner. All of these users have unique profiles and unique IDs that entail them to some level of privileges dependent on their master profile type.

6.6.2 Base64 Encoder Decoder (BED64)

This tool does the basic encoding and decoding of contents. When we need to save content in the Content CRUD Gateway database (described below), the encoder is triggered to encode the binary representation into Base64 text. When the Base64 text is needed in its original format, the BED64 does the decoding and rebuilds the binary into the original media type based on a content profile generated during the encoding stage. Encoding and decoding is an underlying service integrated within other services and tools. Services that save contents in the database or services that display the content on the user interface make use of BED64. If we embed the encoding and decoding process as code in these tools, that would eclipse the benefit of having replaceable and reusable components. Hence this tool is better used as a standalone component (DLL) or as a Web Service that can be called. BED64 has no graphical interface.
We built an application that integrates BED64 DLL and also makes use of Windows built-in Open File dialogue as a proof of concept. The interface shown in Figure 6.32 shows the open file dialogue used to select a file to be encoded.

![Open File Dialogue](image)

Figure 6.32: Open File Dialog Integrated within the BED64 Tool

Figure 6.33 shows the file after encoding.
The menu bar gives us access to Profile Manager and the Content CRUD Gateway. The downloaded or newly created profile gets associated with the content once the encoded objects in saved in the database. The content and the profile create an association that would enable the reversing of the encoding process and the reproduction of the original file contents in their original format, such as an image will be back as an image, etc.

In fact the Base64 Converter tool integrates three reusable components at once. The User and Content Profile Manager, the Content CRUD Gateway, and BED64.
6.6.3 The Content CRUD Gateway (CCG)

This tool supports the **save**, **update**, **retrieve** and **delete** of contents. Since all contents get saved into the database in Base64, these tools integrate the BED64 within it to handle the encoding and decoding portion of the desired CRUD tasks. The tool is one of underlying services that be integrated with other tools. The collection of service established the Service Layer part of the architecture.

![Figure 6.34: CCG Sample Dialog](Image)
When a content to be added to the database, a profile needs to exist for that content or a new profile needs to be created and once that is done all the metadata information is collected and made ready to be shipped with the content when added into the database.

This approach of integrating smaller components to build functional-rich services follows the framework proposed in Chapter Five. This tool has no graphical user interface but Figure 6.34 above shows a prototype that integrates this tool to prove the concept of having it integrated as an underlying service within other tools.

### 6.6.4 Course Browser

This tool allows the end-user to browse through the contents and courses available in the database. If the user is content owner for example, more functionality will be available through the course browser. The content owner will be able to make changes to the content, update its version or decommission it, etc. This tool can integrate with other services to create a more functionally rich environment. This allows the content owner more control, quicker response, and better results from within the one tool. If the user were a student, the user would be able to browse through the courses and see the courses that the user has a record associated with. For example, the user would be able to see a history of courses taken, current courses registered, etc.

If we keep following this approach of abstracting services and integrating them into other tools and services, the resultant infrastructure will become a limitless intelligent and expanding engine.
Figure 6.35 shows a typical output of the course browser. It displays a readable version of the metadata of contents available within the database. This content metadata is searchable and allows one to filter through the contents through various search criteria.

![Course and Content Browser Sample Tool](image)

**Figure 6.35: Course and Content Browser Sample Tool**

### 6.6.5 Chat Window

The chat window serves as a two-way multicast communication tool between the teacher or the TA and the students and also between students themselves. Using this tool,
teaching assistants can conduct their teaching hours. Professors can conduct their office-hours. General question and answer periods, roundtable discussions, and group discussions can also be conducted using this tool. The tool does not support voice and video streaming; however, audio and video streaming can be included if they are implemented as services.

One of the things that this tool allows is session recording and exporting. Messages exchanged can be exported in multiple formats, including Base64 format, which can also be added to the course contents as a course resource or posted on a web page for future references. For example, if roundtable discussion was of value to be added to the course material the session recording will capture the messages exchanged and save the session in the format selected by the user. All off-the-shelf chat tools do not favour latecomers. Users, who join the chat late, miss on the previous part of the chat until someone sends the recorded session via email to them. This may not be acceptable if the time difference is critical such as if the session were two hours before the final exam. By the time the session is sent it will be useless, as the students will be heading to the exam room anyway. Our tool solves this problem.

First of all, the Chat Tools is a Web Service (WS) based two-way multi-user, multi-session, and Instant Messaging software tool. In a WS-based session, all participants refresh their status with the Web Service at preset time intervals. During the refresh, the WS informs any user of any messages waiting for them. These messages are protocol messages that can be used to perform actions on the user's side. More like an email
message waiting for the person in his or her mailbox. When the chat session owner logs in, the session is recorded by default. If user X joins the discussion late, the user can click the "Late Comer" button to refresh the chat session and the user gets the entire message that she missed.

How is this done? The following workflow gives the step-by-step process.

- User X logs in and joins the discussion.
- X clicks the "Late Comer" button
- The chat tool of user X put a WS request on the server for full session text.
- The WS creates a message request and deposits it in the message queue
- On the next "Refresh" by the session owner, the WS traverses the queue and checks if there is any message waiting for the session owner.
  - If there are no messages waiting in the queue for the session owner, the return value for the refresh call is "Free" and it ends here....
  - If the WS finds the request of user X and (in this case) the refresh process returns the value "Messages Waiting" to the session owner
- The Session-owner requests the Messages and the WS returns a collection of messages. In this case, one message says, "send up the recorded session, as user X needs it."
- The Session-owner, sends to the WS the content of the session with the message unique identifier
- The WS creates a message waiting for user X.
• On the next “Refresh” by user X, the WS traverses the queue and checks if there is any message waiting for user X.
  o If there are no messages were waiting in the queue for user X, the return value for the refresh call is “Free” and it ends here....
  o If the WS finds the response of the session owner then “Refresh,” returns the value “Messages Waiting” to user X

• User X requests the Messages from the WS, which is the recorded session content that was sent up by the session owner.

• The Chat displays the content of the chat session on user X’s display.

The Figure 6.36 shows the Main Chat Window tool with login dialog. This chat tool is an application and every application has a profile. When the chat tool is launched it cross references with the server to see which services are up and running and if there is any version better than the local components then it triggers an implicit background download for the new components. All these steps described above are pure Web Service-based message calls.
6.6.6 Web Search Tool

Web searching is a very important aspect of today's research activities. However finding the right resource on the web is not an easy task when it comes to processing the results. Many different web search engines use different algorithms in filtering possible hits. It is time consuming to use multiple search engines to conduct the same search. Also, many of the results will be identical. Other than that, some resources are not available to the general public but only to authorized subscribers. An institute can subscribe to such resources and make the search result available to the students for example through an interface. For example, if an educational institute were to have a membership with the IEEE, an engineering student could use a tool to search the IEEE papers and the resources without having to be a member of IEEE. So, the web search tool farms search
results on multiple search engines through Web Service calls. Of course this is based on the assumption that these search-engines have an open source WS that we integrate with.

Google and Amazon both offer Web Service access to their search engines. Amazon for example, returns information about their books including rating of the book, user feedback, price, etc, in a standardized object-oriented fashion. Integrating all these different Web Services into one search tool would generate a tool that can really saves time.

Integrating Web Services together creates what is called a mesh-up. This mesh-up offers higher levels of abstraction and richness in functionality for our search engines. Using mesh-up, services get integrated together into a virtual service that has a single interface. This single interface is used to interact with all the services implicitly. This mesh-up in itself can be integrated in another mesh-up, and so forth. Following this approach we can build a multi-layer service infrastructure that can be expanded horizontally and vertically. Figure 6.37 shows the conceptual model of the search engine mesh-up.

Figure 6.38 shows the Web Service based mesh-up search engine tool prototype that integrates Google, Yahoo and Amazon into one search tool. It integrates the search capabilities of the three search engines within their contexts. For example, Amazon searches its own database whereas Yahoo and Google search the Internet.
Figure 6.37: Conceptual Model of Search Engine Mesh-Up
The core infrastructure of the proposed system is a service-oriented architecture (SOA) mashup of open source APIs. "A mashup is a website or web application that seamlessly combines content from more than one source into an integrated experience." As a mash-up our system will leverage other open-source and API technologies in a vertical fashion to build a new higher tier of abstraction to achieve a better user experience through ease of use, richness content and added functionality.

Figure 6.38: SOA-Based Web Search Mesh-up

6.6.7 Virtual Classroom Space (VCS)

The virtual classroom space is a shell tool that integrates the various services mentioned so far except for the Contents CRUD Gateway since it not within the context of the classroom. In this virtual classroom a user could launch a chat session, video and audio session, a web search tool, content browser, user profile manager and so forth using the menu bar and the various text entry sections as shown in Figure 6.39.
The virtual classroom is a major container of containers that integrates other services and other tools and facilities within its shell. This simplified collaborations space is extensible due to the fact that its profile is extensible also.

Users, as Figure 6.39 shows, have access to their profile, the current courses registered and other collaboration tools that are accessible through the menu bar. The Virtual Classroom Template has an embedded Web-Service based chat system that used the
exact infrastructure that support the Chat tool described in 6.6.5 above. The user also has access to the “Viewing Area”. The Viewing Area is a web browser embedded within the tools panel. This way that viewing area can hold a media streaming tool such as Microsoft Media Player embedded into an HTML document and launched on the viewing area embedded web browser. Or the view area can hold a PowerPoint Slide show embedded within an HTML document. More importantly it can hold an ASP.NET interactive application, which can be real-time polling of opinions, or answering multiple-choice test, web-based multi-user whiteboard where users can collectively work on the same graph, etc. Having the viewing area as a web browser, ensures that the user can see any media item or multimedia content that can be embedded or displayed within the context of a web browser which means a wide assortment of media type contents.

Also, any tool new that becomes available can be added to the option menu, which enhances the user experience. Whenever the user logs in, the new application profile is downloaded. If a new menu option is available, the content of the menu option is also downloaded and the shell refreshes its menu content and the new option will show and the user will be able to launch this new service.

6.6.8 Main Collaboration Shell (MCS)

The main collaboration shell, as the name suggests, is the shell that can hold all the previous tools, services, and facilities that support collaboration. This shell is a smart client. We discussed smart client technology in length in Chapter 3. In short, smart clients provide the security context of the web browser and the richness of functionality of a
desktop application. Smart client is a Microsoft technology that simplified application
installation and updating. Smart Client applications are downloadable by simply clicking
on a URL. Once the application is downloaded and installed on the client machine it runs
in a safe environment, as it does not need to update the registry or access any directory on
the file system beyond its own temporary location.

Once the smart client application is downloaded and installed on the user’s machine, the
application manifest gets download also. A manifest in the context of smart client
technology is an XML document that contains the names of components downloaded and
their full version ID card plus some other information that the smart client. Each time the
smart client application runs, it connects back with the deployment server where it was
originally downloaded. Once the connection has been established, the smart client
compares its manifest with the one on the deployment server. Once a mismatch between
a component info on the server (or more) and a component info on the user’s machine) is
detected, a download of the new comments is initiated and a local manifest gets updated
to reflect the update-to-date version of the application components. This ensures the
application is always up-to-date and bug fixes or new feature are propagated in a timely
manner. To run the Smart Client Shell application for the first time, the user, for example,
gets an email that contains the URL where the shell Smart Client is located for
deployment. Clicking the link will immediate triggers the download. Figure 6.40 shows
the windows form indicating the download in progress.

![Launching Application](image)

**Figure 6.40: Smart Client Launching Application Download Window**

Figure 6.41 shows the System Warning that an application is about to be installed on the machine; this is a .NET event handler that discovered the application is a Smart Client. The application being installed is the Collaboration Shell.

![Application Install - Security Warning](image)

**Figure 6.41: Smart Client Initial Install Warning Message**

Figure 6.42 shows the collaboration shell first login with its basic toolbar menu. The shell is expandable and remotely controllable based on the application and user profile. Once
the user update its own profile the application profile gets updates automatically to reflect
the new services and tools that the user can have access to.

When the user performs its first login, the basic users’ profile is activated. This activation
triggers an implicit upgrade to the application profile that this user has access to. On
subsequence logins then application profile will be downloaded and consequently the
user will have access to a more extensive toolbar and menu option as clearly shown in
Figure 6.43 below. New options can be added and some options can be disabled based on
the application profile or availability of services at any moment. When the user clicks
any menu bar option, say the Course Browser tool, the application automatically checks
the local cache for the service and if the service is available, cross checks the service
version number with the one residing on the server to ensure that the user has the latest
version of the software tools. If the local service is outdated, then the application will
trigger a service call to the tools repository and request the new copy of the service or
tool downloaded. The application updates its local manifest of services with the latest
version of the downloaded Course Browser tool for subsequent invocations. Every time
the user clicks for a service or tool, this workflow takes place. This ensures the user has
the latest tools and service available. Now if a service is not available for whatever
reason, the application admin can remotely deactivate that menu option by simply
switching off that option in the application’s profile.
6.7 Summary and Conclusion

The various tools that have been implemented and presented within this chapter highlighted the importance of building software system functionality as services and components. We must think in abstraction without losing sight of other tools around that we can integrate with our environment.
Building complex TeleLearning systems out of components and services and capitalizing on the power of other tools and services out there greatly enhances the possibility of realizing the full potential of distance learning. Furthermore, breaking any complex system into components and services allows us to build applications in stages and in parallel; and certainly, permits the update or replacement of services and components in a totally seamless fashion.

It's important to say that these tools can themselves be expanded and integrated with other tools to create a repository of services that can integrated with, and support, other systems.

The tools implemented demonstrate that expandable, flexible, interoperable and component-based frameworks can be built by following pure SOA approach. With the use of application profiles and following component-based design new features and functionality can be added or removed from the Collaboration tool in a matter of minutes if the components are available in the database.
Chapter 7: Conclusions and Future Work

This chapter concludes the description of the development and implementation of the solution to the problem originally hypothesized: how can one utilize current computer and communication technology to create multimedia collaborative spaces.

The chapter presents the final thoughts of what was planned to be implemented and also presents the outcome of the implementation. The contributions that the thesis has made are also briefly highlighted followed by a discussion of the limitations to these contributions. The chapter finally presents suggested future work that can expand and verify the work done within the thesis.

7.1 Discussion

The thesis problem identified two major areas for research. One area was to enhance how multimedia file contents are managed, maintained and delivered. The other area of research was to standardize the process of collaboration both in tools and communication in order to build forward compatible systems that can be expanded and reused.

7.1.1 Designing in Abstraction and Not in Isolation

The findings were supportive of the idea that most collaboration tools are built from the ground up by one vendor without taking into account the various re-useable tools and standards that can be integrated with the vendor’s work. Leopold Reif of Deutsche
Telecom identified that “the strongest niche that differentiates a collaborative environment from another is degree of integration of useful and meaningful software applications in the delivery of the knowledge and for supporting the various stakeholders. Such integration offer enrich to the traditional course. It is what makes the content providers more effective” [110]. It is no longer accepted that we decide how to build in isolation of the context. Reinventing the wheel is counter-productive and opposes the foundation of sound engineering process. Also, standards can no longer be accepted to be the by-product of the development process, but rather standards should take the front seat. Standards must be applied and followed whenever possible. Only then can major integration of various collaboration tools come to fruition.

7.1.2 In TeleLearning Learning Objects Include the Software Tools Used

Packaging the entire collaboration environment into a full-fledged RDBMS provides a unified way of looking at the various parts that make up the collaborative environment. RDBMS also provides built-in services that support business intelligence. In an educational application, the educational institute can produce statistics instantly, for example, on the usability of any software tools, or the access rate for any course contents. Benchmarks and statistics for peak-access performance versus number of concurrent users can be produced in short time using these RDBMS built-in tools. In fact, reports can be generated automatically on predefined schedules or triggered at will. Today’s RDBMS have comprehensive features and are very functionally sophisticated systems. This view is in synchronism with the concept of design in abstraction and not in isolation.
Since RDBMS provide a suite of services that add value to the information provided and produced by any educational institute, it is reasonable that RDBMS be incorporated to the full into the design and implementation of comprehensive collaboration systems. This way the management of the collaborative spaces can switch focus and resources from actually micro managing the content to building better and more useable content. With up-to-the-second reports generated by RDBMS built-in reporting and business intelligence services sound and informed decisions to deprecate and decommission outdated courses, content and tools can be made.

7.2 Contributions

The thesis contribution can be divided into two main groups.

- The first is the streamlined and simplified framework for maintaining and managing multimedia contents. This contribution involves the unification of all media types into one type that is Base64-encoded text. This contribution employs a full-fledged database system for the storage of the media files once they are pure text. These two technologies – the database and the Base64 encoding- allow us to conceptualize a SOA data access layer that handles CRUD operations. The resultant contribution was a SOA multi-tiered framework with database infrastructure for the simplification of management and delivery of multimedia contents.
• The second contribution is a SOA framework for building an expandable and interoperable multimedia collaborative space. The framework followed the basic definition of what constitutes a service in a pure SOA. A service is a well-qualified functionality that can be used as standalone without the influence of the context or the content of other functionality or service. This allows us to identify unique services that fuse together into a collaboration environment. The framework was followed with illustrative prototypes to prove the various concepts presented. The prototypes were implementations of services and tools that support a TeleLearning environment. Part of this contribution also was the definition of the virtual multimedia distance-learning space. The definition that was contributed is: "The online distance learning (TeleLearning) environment is a virtual collaborative space that utilizes technology to emulate and improve the experience of the learner in real world."

7.2.1 Contributions Strengths and Opportunities

There are three main strength factors within the proposed solution.

• The proposed framework is a multi-tier SOA model that leverages major cutting-edge technologies in an unprecedented way to streamline the management, sharing and storage of the multimedia contents that are exchanged in a typical collaborative space setting.
• The concept of a database that holds both the software and the contents of a multimedia environment is a unique idea. If an educational institute decided to replicate its collaboration environment including the contents all that is required will be is a database server replication of the entire structure of the original database to new database server. In fact, current database technology can create mirrors of the database contents in real-time, which offers a second line of defence in case of emergencies. Moreover, when databases in other regions are not busy some data access requests could be routed to the low traffic service seamlessly to the end-user. This is a huge advantage over old-fashioned file server backups.

• A future-proof solution to any new multimedia files types. Base64 encoding is binary to text transformation. Since no matter what the new media file type and extension may be, it will be saved in binary on the file system and encoding it to Base64 will make it possible to use the current tools developed to support the current types to support the new type.

7.2.2 Contributions Risks and Weaknesses

A risk and a weakness can be found in the contributions above.

• The risk lies in using RDBMS as the data store for software tools. This is a brand new paradigm and hence it entails the risk associated with being a first in inventing and applying a new concept.
• The weakness of the proposed contribution lies also in some weaknesses of the pure SOA paradigm. Without a doubt, some services will become a bottleneck for service requests and responses. Simply benchmarking throughput may not be sufficient enough to cover the consideration of sudden surge in service requests. This weakness is eclipsed and in fact handled also by the many strengths of SOA. To handle system bottlenecks and the single point of failure situation, SOA suggests the creation of a proxy service dispatcher that multiplexes requests coming from end-use to backend services.

7.2.3 Contribution Limitation

This study has thee limitations.

• The first limitation is the technology type limitation for the prototypes. The study and proposed solution was limited to applying one technology type, namely Microsoft products (.NET and SQL server 2005) and hence further investigation of technologies that run on different operating systems is required.

• The second limitation is hardware and software limitation. The framework is also limited by the hardware and software limitation and requirement such as the hardware requirement of running a real broadband multimedia service.
• The third limitation is bandwidth limitation. The study assumes the end-users would have network access rate at the very least an ISDN and upward of 512 Kbps with 512 MB of RAM and some Gigabytes of free disk space.

7.3 Conclusion

It has been shown that it was possible to design and prototype an SOA multimedia collaborative space using current technology. The process involves the deployment of an innovative framework for the management of multimedia contents by utilizing cutting edge RDBMS service and Base64 encoding. A multi-tiered SOA framework with a database as core infrastructure is the result of the solution to the thesis problem.

Services, technologies and standards have matured enough both at the software and hardware level to enable the building of extensible and reusable systems that are forward compatible and future-proof. Capitalizing on the emerging technology for building more useful and interoperable collaborative space should be pursuit of the highest priority from all interested parties.

7.4 Future Work

The author has seven suggestions for future work.

1. The proposed framework be validated using other technologies and other OS systems such as Java on UNIX, since the prototypes were implemented using pure Microsoft technologies, namely. NET 2.0 and SQL server 2005.
2. The prototype implementation of video and audio multicast services. The implemented prototypes within the thesis did not attempt to implement either of the services.

3. By means of implementation, the framework is to be validated using thin clients, which are web-based applications that can link to the SOA infrastructure.

4. By means of implementation, the framework is to be validated using thick clients, which are desktop applications that can link to the SOA infrastructure.

5. A detailed UML modeling of the workflow of the framework.

6. The design, implementation and integration of non-collaboration services, such as business office functions and registrarial services (e.g., final grades records) into the collaboration main shell to expand the proof of concept.
References


[18] From Wikipedia, the free encyclopaedia,
http://en.wikipedia.org/wiki/WebCT#Background

[19] From Wikipedia, the free encyclopaedia,
http://www.webct.com/service/viewcontentframe?contentID=20154050

[20] Microsoft Office Communicator 2005 Overview, Microsoft Corp. 2005,
http://www.microsoft.com/office/communicator/prodinfo/overview.mspx

[21] Harris-Jones, C., "Office Communicator - The Next Collaboration Tool from

School Press.

[23] Besen, S., and Farrel, J. "Choosing How to Compete: Strategy and Tactics in

Society Standards Activities Board 7-8 March 2006


[27] The World Wide Web Consortium (W3C), http://www.w3.org/

http://www.stanford.edu/class/symbsys205/Diffusion%20of%20Innovations.htm


[32] IEEE SCORM-Sim Study Group,
http://www.ndia.org/Content/NavigationMenu/Meetings_and_Events/Schedule_of_Events/Events/61D0/61D0_IEEE.pdf

[33] Friedman, T, "The World is Flat, A Brief History of the Twenty-first Century",
Published by Farrar, Straus and Giroux April 2005, P 6, 120.

[35] Channel9 Wiki,


http://www.internetworldstats.com/stats.htm

[38] From Wikipedia, the free encyclopaedia http://en.wikipedia.org/wiki/IPv4

[39] From Wikipedia, the free encyclopaedia
http://en.wikipedia.org/wiki/Network_address_translation


[42] From Wikipedia, the free encyclopaedia


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.

http://www.internetworldstats.com/stats.htm


[51] Farber, D., "2006: The year of dual core, virtualization, WiMax" ZDNet Forum,

[52] http://www.gartner.com/research/fellows/asset_94007_1175.jsp

http://www.businessweek.com/magazine/content/05_16/b3929120_mz018.htm
[54] Hyper-Threading Technology, Intel Inc.
http://www.intel.com/technology/hyperthread/

[55] Understanding Multi-Core Processor Architecture, Intel Inc.


[57] Wilcox, J. "Notebook prices going into free fall" Dec., 2000,


[59] Microsoft Visual Studio 2005 Developer Centre, Microsoft,
http://msdn.microsoft.com/vstudio/

[60] Whinston, A., "Interoperability in Global Electronic Commerce", The University of Texas at Austin, 1999

[61] Anderson, R., "The End of DLL Hell", Microsoft Corporation, 2000,
[62] From Wikipedia, the free encyclopaedia,
http://en.wikipedia.org/wiki/Convergence#Mathematics

[63] International Organization for Standardization,


[66] Fiswidgets Client / Server Glossary
http://grommit.lrde.pitt.edu/fiswidgets/fisdocs/cs_glossary.html

http://www.standishgroup.com/sample_research/chaos_1994_1.php

[68] Perks, M., "Best practices for software development projects",
IBM Inc., 2006, http://www-
128.ibm.com/developerworks/websphere/library/techarticles/0306_perks/perks2.html

[69] From Wikipedia, the free encyclopaedia,
[70] "Patterns & Practices: Enterprise Library", Gotdotnet, (2005),

[71] From Wikipedia, the free encyclopaedia,
http://en.wikipedia.org/wiki/Agile_software_development


[74] Pollice, G., "A look at aspect-oriented programming", Professor of Practice,


[76] Mitchell, S., "An Introduction to the Microsoft Enterprise Library", 4guysfromrolla,
(2005), http://aspnet.4guysfromrolla.com/articles/022305-1.aspx

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
[77] Shared Source Initiative, Microsoft,
http://www.microsoft.com/resources/sharedsource/default.mspx


[84] BEA, IBM, Microsoft, SAP and Sun, "Web Services Addressing Submit WS-Addressing Specification to W3C for standardization", August, 2004

[85] From Wikipedia, the free encyclopaedia,


[87] “IDC Reveals Steady Growth for Worldwide RDBMS Market as Top 5 Vendors Hold Their Positions “, IDC, May 2006,
http://www.idc.com/getdoc.jsp?containerId=prUS20180306

[88] Digital Media File Types: Survey of Common Formats, NIST,
http://www.itl.nist.gov/div895/isis/filetypes.html


[91] “Managing the Business of Database Design “, AllFusion,
[92] From Wikipedia, the free encyclopaedia,
http://en.wikipedia.org/wiki/Filename_extension

[93] “Definitions of Broadband on the Web”,
http://www.google.ca/search?hl=en&lr=&rls=GGLJ,GGLJ:2006-
11,GGLJ:en&defl=en&q=define:Broadband&sa=X&oi=glossary_definition&ct=title

[94] “Collaborative Space”, COMP@CT, a European Educational Network, 2006,
http://www.eduprojects.net/sig1/collaborative_space.html


9-7AE3-4942-B466-CC778A3BAB34

[97] “SOA-Based Services Gain Foothold in U.S. Companies' Overall IT and Business Strategy, and Spending Plans in SOA Rise, IDC Finds “, June, 2006,
http://www.idc.com/getdoc.jsp?containerId=prUS20224606

http://webmethods.com/meta/default(folder/0000008660
http://infomotions.com/musings/waves/clientservercomputing.html

[100] From Wikipedia, the free encyclopaedia, 


[103] From Wikipedia, the free encyclopaedia, 
http://en.wikipedia.org/wiki/Base64#MIME


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
[107] “Smart Clients: Combining the Power of the PC with the Reach of the Web”,

D. Straub and J.I. DeGross, editors, 7th Americas Conference on Information Systems
(AMCIS 2001), pp. 216-222, 2001

[109] From Wikipedia, the free encyclopaedia,
http://en.wikipedia.org/wiki/Initialization_file

[110] Reif, L. "Service for Learning", Deutsche Telecom. 1999,