

**ASSEMBLE TO ORDER LEARNING MANAGEMENT
SYSTEM**

by

Baljeet Singh Dhaliwal

Bachelor of Engineering (Mechanical Engineering), Calcutta University, 1992

THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
in the School
of
Interactive Arts and Technology

© Baljeet Singh Dhaliwal 2006
SIMON FRASER UNIVERSITY
Summer 2006

All rights reserved. This work may not be
reproduced in whole or in part, by photocopy
or other means, without the permission of the author.

APPROVAL

Name: Baljeet Singh Dhaliwal
Degree: Master of Science
Title of Thesis: ASSEMBLE TO ORDER LEARNING MANAGEMENT SYSTEM

Examining Committee: Dr John Bowes,
Chair

Dr Vive Kumar, Senior Supervisor,
Assistant Professor

Dr John Nesbit, Supervisor,
Associate Professor

Dr Belgacem Ben Youssef , External Examiner,
Assistant Professor,
Simon Fraser University

Date Approved: _____



**SIMON FRASER
UNIVERSITY** library

DECLARATION OF PARTIAL COPYRIGHT LICENCE

The author, whose copyright is declared on the title page of this work, has granted to Simon Fraser University the right to lend this thesis, project or extended essay to users of the Simon Fraser University Library, and to make partial or single copies only for such users or in response to a request from the library of any other university, or other educational institution, on its own behalf or for one of its users.

The author has further granted permission to Simon Fraser University to keep or make a digital copy for use in its circulating collection, and, without changing the content, to translate the thesis/project or extended essays, if technically possible, to any medium or format for the purpose of preservation of the digital work.

The author has further agreed that permission for multiple copying of this work for scholarly purposes may be granted by either the author or the Dean of Graduate Studies.

It is understood that copying or publication of this work for financial gain shall not be allowed without the author's written permission.

Permission for public performance, or limited permission for private scholarly use, of any multimedia materials forming part of this work, may have been granted by the author. This information may be found on the separately catalogued multimedia material and in the signed Partial Copyright Licence.

The original Partial Copyright Licence attesting to these terms, and signed by this author, may be found in the original bound copy of this work, retained in the Simon Fraser University Archive.

Simon Fraser University Library
Burnaby, BC, Canada

Abstract

Every Learning Management System (LMS) comes with a unique bundle of functionalities. Most of these functionalities are neither portable nor common across various LMS. That is, a functionality developed for one LMS cannot be integrated within another LMS. Having noted that the process of software development is not very different from industrial production, we suggest a way of applying the principles of production management to software development in eLearning, particularly in the design of LMS. The functionalities in LMS are developed based on the notion of Assemble to Order (ATO) commonly found in industrial production. The thesis contends that LMS designed using the ATO approach will allow scalable and adaptable integration of functionalities across LMS. The thesis substantiates this claim using a case scenario and analytical results obtained from software engineering metrics such as Function Point Analysis (FPA) and Lines Of Code (LOC).

Keywords:

Learning Management System; eLearning; Education Technology; Interoperability; ATO software development.

To my family and friends!

Acknowledgments

I would like to thank my senior supervisor Dr. Vive Kumar, for his supervision on this thesis and throughout my graduate studies. He has been a thorough professional and an excellent guide during my entire course of interaction with him.

I also want to thank Dr. John Nesbit, who has always extended his guidance, first as my manager in Education, Technology & Learning (ETL) department of Technical University of British Columbia (TechBC), then as one of my instructors during my graduation and finally as my supervisor in completing this research thesis.

I further want to thank Chris Groeneboer for her support and guidance as my manager and a friend and for guiding me on software metrics and productivity measurements.

Finally, I would like to thank Dennis Humphrey, my current manager at SFU and also my other colleagues, friends and family without whose support I would not have been able to complete this thesis.

Abbreviations

API: Application Programming Interface
APICS: American Production and Inventory Controlled Society
ATO: Assemble To Order
CAD: Computer Aided Design
CAM: Computer Aided Manufacturing
CLE: Collaborative Learning Environment
CMS: Course Management System
CRM: Customer Relations Management
CRUD: Create Retrieve Update Delete
EI: External Input
EIF: External Interface File
EO: External Output
EQ: External Inquiry
ERP: Enterprise Resource Planning
ETO: Engineer To Order
FP: Function Point
GUI: Graphical User Interface
HTML: Hypertext Markup Language
IFPUG: International Function Point Users Group
ILF: Internal Logical File
IT: Information Technology
JAXP: Java API for XML Parsing
LMS: Learning Management System
LOC: Lines Of Code

MIT: Massachusetts Institute of Technology
MIT: Massachusetts Institute of Technology
MPS: Master Production Schedule
MRPII: Manufacturing Resource Planning
MTO: Make To Order
MTS: Make To Stock
OKI: Open Knowledge Initiative
OSID: OKI Services Interface Definition
SAKAI: Synchronized Architecture of Knowledge Acquisition Infrastructure
SAX: Simple API for XML
SFU: Simon Fraser University
SKU: Stock Keeping Unit
TPP: Sakai Tools Portability Profile
UAF: Unadjusted Function Point
W3C: World Wide Web Consortium
WANE: Workforce Ageing In The New Economy
XML: eXtensible Markup Language
XSLT: eXtensible Stylesheet Language Transformation

Contents

Approval	ii
Abstract	iii
Dedication	iv
Acknowledgments	v
Abbreviations	vi
Contents	viii
List of Tables	xii
List of Figures	xiii
1 Introduction	1
1.1 Manufacturing and IT	1
1.2 SAKAI	2
1.3 Software Development as a Manufacturing Process	4
2 LITERATURE REVIEW	7
2.1 Online Learning	7
2.1.1 Learning Management System (LMS)	8
2.2 Assemble To Order (ATO)	10
2.2.1 Master Production Schedule	11
2.2.2 Make To Stock (MTS)	11

2.2.3	Assemble To Order (ATO)	13
2.2.4	Make To Order (MTO)	13
2.2.5	Engineer To Order (ETO)	14
2.2.6	Customer Order Decoupling Point	14
2.2.7	Manufacturing Environments and MPS	14
2.2.8	Detailed Analysis of Assemble To Order Environment	16
2.2.9	Advantages of ATO	18
2.3	Software Reusability	18
2.3.1	Software Reusability and LMS	19
2.3.2	OSID(OKI Services Interface Definition)	22
2.3.3	SAKAI	22
2.4	Open Source/Free Technologies	24
2.4.1	eXtensible Markup Language (XML)	24
2.4.2	eXtensible Stylesheet Language Transformation (XSLT)	25
2.4.3	Java™ API for XML Processing (JAXP)	25
3	METHOD	27
3.1	Typical Request Response System	27
3.2	ATO Request Response System	29
3.2.1	The Design for ATO LMS	30
3.2.2	Fixed Part	30
3.2.3	Variable Part	32
3.3	Evaluation	32
3.3.1	Event Sign-up Tool (ESUT)	33
3.4	ATO Source Codes	37
3.4.1	ATO-ESUT-basecode	37
3.4.2	Package esutfixed	38
3.4.3	Package esutvariable	39
3.4.4	XSLT Documents	40
3.4.5	Package javaxslt	41
3.4.6	XML Schemas	44
3.4.7	CMS-ESUT	46
3.4.8	Eledge-ESUT	47

3.4.9	EduTools Evaluation	48
4	DISCUSSION	49
4.1	Code Sizing Metrics	49
4.2	LOC Vs FP	54
4.3	Event Sign-Up Tool	55
4.3.1	Database Structure Comparison	55
4.3.2	LOC Comparison	58
4.3.3	Function Point Comparison	58
4.4	EduTools	61
4.4.1	File Exchange Tool	63
4.4.2	Bookmarks	64
4.4.3	Self Assessment Tool	66
4.4.4	EduTools FP Analysis	71
5	CONCLUSION, LIMITATIONS AND FURTHER RESEARCH	72
5.1	Limitations	73
5.2	Further Research	73
A	XML Schemas and XSLT Files	74
A.1	XML Schemas	74
A.1.1	DeleteEvent	74
A.1.2	ListEvents	75
A.1.3	NewEvent	77
A.1.4	NewSession	78
A.1.5	PostEvent	79
A.1.6	ReEnterEventData	80
A.1.7	SaveEvent	82
A.1.8	SaveSession	83
A.2	XSLT Files	84
A.2.1	createEvent	85
A.2.2	createSession	87
A.2.3	listEvents	89
A.2.4	reEnterEventData	93

A.2.5	reEnterSessionData	96
B	Source Code	100
B.1	ATO-ESUT-basecode	100
B.1.1	Package ca.sfu.elinc.esutfixed	100
B.1.2	Package ca.sfu.elinc.esutvariable	139
B.1.3	Package ca.sfu.elinc.javaxslt	145
B.2	CMS-ESUT	153
B.2.1	Package ca.sfu.elinc.esutfixed	153
B.2.2	Package ca.sfu.elinc.esutvariable	153
B.2.3	Package ca.sfu.elinc.javaxslt	170
B.3	Eledge-ESUT	170
B.3.1	Package ca.sfu.elinc.esutfixed	170
B.3.2	Package ca.sfu.elinc.esutvariable	170
B.3.3	Package ca.sfu.elinc.javaxslt	189
C	Ant Build File and CSS	190
C.1	build.xml	190
C.2	stylesheet.css	193
	Bibliography	199

List of Tables

4.1	UAF Calculation Table	53
4.2	EI Rating Table	54
4.3	LOC Comparison Table	59
4.4	EduTools FP Comparison Table	71

List of Figures

2.1	MPS in Production [51], by permission	12
2.2	Customer Order Decoupling Point (CODP)	15
2.3	Manufacturing Environments	16
2.4	OKI Architecture [55], by permission	21
2.5	SAKAI Framework [17], by permission	23
2.6	XSLT Transformation [9], by permission	25
3.1	Typical Request Response System	28
3.2	ATO Query System	29
3.3	ATO Update System	31
3.4	Create Event Page	33
3.5	Create Session Page	34
3.6	ESUT Main Page	36
3.7	Re-enter Event Data	42
3.8	Re-enter Session Data	43
4.1	Functionality recognized in Function Point counting [19], by permission	51
4.2	Eledge Events Table	56
4.3	Eledge Sessions Table	56
4.4	CMS esut_course_events Table	57
4.5	CMS esut_course_sessions Table	57
4.6	LOC Comparison Chart	60
4.7	FP Comparison Chart	62
4.8	Function Point Comparison for EduTools Applications	71

Chapter 1

Introduction

1.1 Manufacturing and IT

The current Information Economy is “an economy based on the exchange of knowledge information and services rather than physical goods and services” [20]. We are all aware of the growth of the information technology industry since the introduction of the world wide web in the early 90s. The growth is anticipated to continue for the next 5-10 years. According to the Bureau of Labor Statistics (BLS), among the professional and related occupations computer and mathematical occupations are the fastest growing occupations and are projected to add approximately 2,000,000 jobs between 2000 and 2010. Ten of the 30 fastest growing occupations are computer-related [31]. As per WANE¹, despite the global down-turn of the IT industry in 2001-2002, in all of the countries covered by WANE study, IT industries are doing well and are growing at a stronger rate. However, no matter how strong the growth has been, the software development industry has existed only for a few decades and hence our experience in software development is relatively limited.

Manufacturing, on the contrary, has existed almost since the beginning of human commerce. Everything we see around us is manufactured or is related to manufacturing. Be it the computers, or the furniture, or the buildings, all of them either are manufactured, or use parts that are manufactured. The modern manufacturing industry as it exists today

¹Workforce Ageing In The New Economy (WANE) is an international comparative study of information technology employment. Funded by the Social Sciences and Humanities Research Council of Canada (SSHRC) and global partners, the project involves a consortium of academic researchers and industry partners in Australia, Canada, the European Union, and the United States

came into being due to the strong impetus provided by the advent of Industrial Revolution in the 18th century. “The Industrial Revolution may be defined as the application of power-driven machinery to manufacturing” [22]. The steam engine, developed in 1769, initiated the beginning of the first Industrial Revolution that implied the transition from an agriculture based economy to an industrial economy in Britain [54]. Manufacturing, since then, has come a long way. We now have a mature manufacturing industry that is more than two centuries old, with robust experience in planning and execution. The growth of the manufacturing industry has been further fueled by the introduction of IT in manufacturing organizations. The advent of CAD/CAM, ERP, and CRM² systems has further added to the speed at which manufacturing plans can be executed.

Manufacturing is not the only industry that has benefited from the growth of IT. Almost every industry has undergone a considerable change due to the strong progress made by IT in the past decade. Education is no exception. The spate of Learning Management Systems is a direct result of the impact of IT in Education. At present, there are more than 50 Learning Management Systems³ that are available in the market. Some of them are proprietary and others are open source. Almost every reputable university and educational institution in the world has used one or more LMS. This clearly indicates that eLearning is a phenomenon that is poised to stay with us, notwithstanding the form and the extent.

1.2 SAKAI

One such open source learning management system is SAKAI - Synchronized Architecture of Knowledge Acquisition Infrastructure. “The SAKAI Project is a community source software development effort to design, build and deploy a new Collaboration and Learning Environment (CLE) for higher education” [23]. SAKAI Project is a collaborative effort by Indiana University, University of Michigan, Stanford University, Massachusetts Institute of Technology (MIT), uPortal Consortium, and Open Knowledge Initiative (OKI) by MIT. The project was started in January 2004 and has released its version 2.1.2 in April 2006. At the core of SAKAI, it extends the concept initialized by OKI (Open Knowledge Initiative).

²CAD/CAM: Computer Aided Design/Manufacturing, ERP: Enterprise Resource Planning, CRM: Customer Relation Management

³as listed in <http://www.EduTools.info/course/compare/byproduct/index.jsp>

OKI is an initiative started by MIT to build standard services used by educational institutions for developing interoperable applications. “The Open Knowledge Initiative develops specifications that describe how the components of an educational software environment communicate with each other and with other enterprise systems” [48]. OKI suggests the standardized way of implementation of services so that the educational institutions developing applications with OKI specifications, can make use of common services between applications and can re-use the services, already existing. OKI uses standards such as IMS Global Learning Consortium [15] as guidelines for developing the architecture behind OKI. In doing so, OKI promises the blend of legacy educational applications, current developments and next generation eLearning software to co-exist. In order to achieve their objectives, OKI has introduced a series of Open Service Interface Definitions (OSID) to implement core functionalities of services needed by an educational institution.

As mentioned earlier, OSID [25] are a series of services developed by the OKI team to provide guidelines to educational institutions and other LMS contenders, for developing interoperable applications. As per OKI website [48], “the OSID are an abstraction layer between the programmer and the enterprise infrastructure systems of his or her campus. Each OSID is characterized by a tightly defined set of methods and strict boundaries”. Some of the OSID developed by OKI team are Agent, Assessment, Authentication, Dictionary, and Course Management. Each of these OSID provides information about the features that need to be implemented for applications to be interoperable.

SAKAI has released its own set of specifications called SAKAI Technology Portability Profile (TPP), which besides including other technologies such as JavaTM Server Faces, Hibernate, Spring, and JetSpeed, is using extension of OKI-OSID for services. SAKAI has developed its own set of services which are built on top of OKI-OSID. The primary role of these services is to share themselves among various applications. Some of these services are EmailService, AnnouncementService, AssignmentService, UsageSessionService, and so on.

The design processes that have been driving the development of SAKAI are in harmony with the proven and tested processes that currently drive the manufacturing industry. The next section elaborates on how to tap into the lessons learned from mature manufacturing processes and apply them in the next-generation software development.

1.3 Software Development as a Manufacturing Process

Our experience in software development is limited as compared to our experience in manufacturing environments. However, the process of software development is not very different from Industrial Production [59]. It is definitely possible for us to make use of knowledge from manufacturing industries to improve software development. Manufacturing organizes into four strategic categories depending upon the inventory it needs to stock to be able to cater to customer needs at minimum supply lead time⁴. The categories are Engineer-To-Order (ETO), Make-To-Order(MTO), Make-To-Stock(MTS) and Assemble-To-Order(ATO).

In an ETO environment, the products are engineered and developed. In this kind of environment, not even raw materials are inventoried because each product is unique.

In an MTO environment, the supplier has the design of the final product and the execution of the project starts after receipt of the order. Raw material is inventoried either in anticipation of an order or of material which is common among multiple products.

In an MTS environment, the finished products are inventoried and the products are shipped as soon as the orders are received.

In an ATO environment, the stocks are inventoried at semi-finished assemblies or finished parts level and can be used to serve wide range of finished products by using various permutations and combinations of semi-finished assemblies or finished parts.

The advantages of developing software with ATO approach are as follows:

- No organization or industry has unlimited resources. ATO design implementation enhances the learning experience by easily migrating a viable concept in one LMS to another with minimum efforts, thereby extending the success of one institution to another.
- Reduces the cost of software development by reducing the development and designing efforts.
- Provides better stability as most parts of the code are already tested.
- Provides better security in the long run as the system is implemented in various LMS and hence gets tested for various scenarios as against only one.

⁴Supply Lead time is defined as time required to supply an item to its end user or customer [28]

No two LMS are identical because every LMS is build on unique database schema. Hence, it is not possible to have an MTS approach for applications in LMS. In software development, MTS approach is used for stand-alone products. The applications in LMS are developed using either MTO or ETO approach. The applications are either designed from the concept level (ETO) or sometimes the concept is copied from an existing LMS (MTO) or some other source. Both MTO and ETO approaches have higher supply lead times when compared to ATO approach. Thus, any application developed using ATO approach will improve on development time for migrating it into another LMS.

This thesis proposes the application of ATO practices observed in the manufacturing industry in software development, specifically in educational environments. We suggest that the software developed in eLearning environments be developed in such a way that they can be easily integrated into LMS or other eLearning components with minimum effort.

The OKI-OSID or SAKAI-OKI-OSID, though founded on similar philosophies, do not give enough scope to move applications from one LMS to another. For two applications to be OKI-OSID interoperable, both of them have to implement same series of OSID services. It is thus advisable to implement standard set of OSID provided by the core OKI team. However, even then, the interoperability is limited to the extent of other LMS's implementing standard OKI-OSID. Any customized version of OSID, such as the SAKAI-OSID, provides even less interoperability with the LMS running different set of OSID. For example: SAKAI applications are not portable to any other LMS, unless the LMS is built on SAKAI OSID which basically makes it another instance of SAKAI.

The thesis provides an XML, XSLT and JAXP based solution for enhancing the interoperability of applications among various LMS. Further, the database schema plays a greater role in making applications interoperable. Most of the eLearning applications are database driven, using the basic CRUD⁵ database operations. No two educational institutions would have similar databases unless they implement same LMS. In general, database operations are tightly integrated into the applications making it practically impossible for the application to be interoperable among LMS. At the same time, it is not practical for educational institutions to migrate to any one kind of database schema. So, for an application to be interoperable this thesis suggests a solution that would reduce the database dependency of an application.

⁵Create, Retrieve, Update, Delete

This can be achieved by developing ATO-based applications using XML schemas instead of directly talking to backend databases. Any application developed using this approach would use XML schema as input for “R” of CRUD and would generate XML schema output for “C”, “U”, and “D” portions of CRUD. If an ATO approach is used while developing the applications, the applications can be integrated with other LMS quite easily. The XML Schema thus becomes a middle layer between the database and the application. The only non-reusable portion of the application will be Input XML creators and Output XML processors. These would compare directly with the ATO in manufacturing, where most of the time even after receiving the order, some processing needs to be done, thereby increasing their lead time as compared to the MTS scenario.

The ATO-LMS extends the idea of ATO from manufacturing to eLearning. The ATO approach helps in sharing of applications among various LMS with relative ease. A sample application called Event Sign-up Tool is developed to demonstrate the advantages of ATO over other approaches. A basecode for the application is developed, and is then extended to two different LMS, viz; CMS of Simon Fraser University and Eledge of University of Utah. The efforts needed to re-develop the application in the LMS is calculated using Lines of Code Matrix and Function Point Analysis. It is observed that ATO approach reduces the redevelopment effort by more than 70% when compared with MTO or ETO approach. To further support the research, i.e; the advantage of ATO approach over traditional ETO or MTO approach, the FP analysis is carried out on three more applications and the result observed is in line with that observed for Event Sign-up Tool.

Chapter 2

LITERATURE REVIEW

Extensive work is carried out in fields of both eLearning and Production Management. It is imperative to understand some important concepts of both eLearning and Production Management before delving into detailed design discussions and analysis of ATO-LMS and our experiment. Some of the important concepts are discussed below.

2.1 Online Learning

The Internet gave birth to online learning, also known as eLearning. “eLearning is the continuous assimilation of knowledge and skills by adults stimulated by synchronous and asynchronous learning events - and sometimes Knowledge Management concepts - which are authored, delivered, engaged with, supported, and administered using Internet technologies” [44].

Online learning started with students interacting with their instructors over email, instant messengers and by submitting of their assignments as attachments to Learning Management Systems, which not only host the course content, but also contain class lectures in audible/video media, discussion boards, team management, learning object repositories, and so on. It gives both the students and the instructors flexibility to access the information at their convenience.

“Learners who use the Internet for learning purposes have reported a greater engagement in the learning experience than in the case of the more static learning associated with the traditional classroom. Online learning enables the instructor to monitor the learners progress continuously. Learners become involved in the learning process and modules can be designed

to suit different learning styles” [18]. The use of the Internet for educational purposes is not limited to Universities or higher education institutions alone, it is extensively used by high school students as well. According to a report by Pew Internet and American Life Project [32], the potential role of Internet as an educational tool is often the primary reasons for families to apply for Internet connection. The report stated that three-quarters of online teens said that the Internet helped them do their schoolwork.

The concept of eLearning is not restricted to educational institutions alone. According to a report published by the Conference Board of Canada eLearning at workplace can be viewed as a means of delivering three key outcomes: improved and consistent rates of lifelong learning, improved productivity, and improved innovation and competitiveness [45]. eLearn magazine conducted a survey [36] on use of online learning in a workplace and found that 25 percent of the respondents to the survey indicated that in 2004 eLearning was already the dominant form of training in their organization, while another 50 percent predicted that eLearning would become the dominant form of training within their organization by 2010.

While there are many modes of delivery for eLearning, LMS is becoming a preferred and predominant mode of delivery for education and training in institutions of higher education, in high schools, and in the industry.

2.1.1 Learning Management System (LMS)

The initial use of internet for educational purposes gave birth to the idea of Learning Management Systems. LMS are conceived as repositories of learning experiences where institutions can organize the learning rather than just exchanging emails or submitting assignments on emails. According to Brandon Hall [21], “A Learning Management System, or LMS, is software that automates the administration of training events. All LMS manage the log-in and registration of users, manage course catalogs, record data from learners, and provide reports to management”. An LMS can be described as a nerve system of a total eLearning solution whether the organization is using computer-based training (CBT), web-based training (WBT), document-based training (DBT), instructor-led training (ILT), or blended training methods (BTM) [8].

There is hardly any higher education institution in Canada, United States of America, and other developed countries that does not use at least one LMS. Most of them are using more than one LMS. An LMS survey by ASTD (American Society of Training and Development) conducted in 2005 [12] exemplified the overwhelming success of LMS in higher institutions.

Out of the 153 responses to the survey, 81.7 percent indicated to have been using an LMS. The survey further showed that around 55.5 percent of the institutions are either very satisfied or satisfied with their LMS, where unsatisfied and very unsatisfied contributed to 14.2 percent. While it was difficult to find out the exact percentage for reasons for selecting an LMS (as users were allowed to pick more than one reason), the majority felt that it was to centralize the management activities centered around learning.

LMS can be broadly classified into

- Proprietary Learning Management Systems
- Open Source Learning Management Systems

As the name suggests, Proprietary Learning Management Systems are owned by a group of stakeholders, that can range from an individual to a publicly listed company. There is a price associated with purchase of the software and annual maintenance contracts. The source code in most of these cases is confidential and the end-user organization has to depend upon the supplier for any major customization of the source code. The end-user is sometimes allowed to customize to a certain extent, restricted by the boundaries of the customization software needed for such changes. Two of the major proprietary learning managements systems are WebCT and Blackboard. The advantages and disadvantages of using WebCT and Blackboard as compared to open source Learning Management Systems are in line with advantages and disadvantages between commercial software and open source software. Both WebCT and Blackboard provide their own customization modules through PowerLinks [50] and Building Blocks[®] [7] respectively. Both PowerLinks and Building Blocks[®] are customization tools needed to develop applications for WebCT and Blackboard only. Both the systems provide their end-users with a flexibility to develop their specific applications. However, an institution implementing both the LMS will have to develop the whole application from start for each of the LMS, which is equivalent to developing the same application twice with no advantage of reusing the code.

Open source learning management systems are freely available Learning Management Systems, mostly governed under GNU General Public License [40]. These are not-for-profit software which can be downloaded by any individual or an organization and can be modified to their requirements. “Recent trends indicate that governmental and not-for-profit organizations seem to be adopting open source software at a faster rate than any other business or consumer segment” [4].

There are many reasons for adopting an open source Learning Management System and one of the obvious reasons is cost associated with a commercial LMS. “But cost is not the only or even the prime reason to look beyond available proprietary LMS solutions. The ability to modify software is an important consideration for many institutions that need to address specific teaching and learning requirements. Others need to integrate a new LMS with existing systems” [11].

Like proprietary LMS, there are many open-source learning managements systems available in the market. Some of the most popular ones are Moodle [43] and SAKAI [23]. Moodle is primarily developed in Linux using Apache, MySQL and PHP (the combination of these technologies is also known as the LAMP platform), but is also regularly tested with PostgreSQL and on Windows XP, Mac OS X and Netware 6 operating systems. SAKAI on the other hand is “a community source software development effort to design, build and deploy a new Collaboration and Learning Environment (CLE) for higher education. The Project began in January, 2004” [23]. As mentioned earlier, SAKAI Project is a collaborative efforts by Indiana University, University of Michigan, Stanford University, Massachusetts Institute of Technology (MIT), uPortal Consortium and Open Knowledge Initiative (OKI) initiative by MIT.

Despite there being several open source LMS, not much effort has been noted in regard to reusability of applications or software code among different LMS, even though there might be several LMS running on similar platform, be it JavaTM, or PHP or .NET based. For example, although both SAKAI and Eledge [47] run on JavaTM platforms, it is practically impossible to share applications between these two LMS.

2.2 Assemble To Order (ATO)

Manufacturing environments (also called production strategies) are broadly classified into

- Make To Stock(MTS) environment
- Assemble To Order (ATO) environment
- Make To Order (MTO) environment
- Engineer To Order (ETO) environment

A production house selects an environment for its product line on the basis of choosing the Master Production Schedule (MPS) approach [52].

2.2.1 Master Production Schedule

MPS is rightly described as Master of All Schedules [51]. MPS is the pivotal point in manufacturing business when demand from marketplace is balanced with the capabilities and capacities of the company and its suppliers in real-time terms. As per APICS [2], MPS is an “anticipated build schedule for those items assigned to the master schedule. The master scheduler maintains this schedule, and in turn, it becomes a set of planning numbers that drives material requirements planning. It represents what the company plans to produce expressed in specific configurations, quantities, and dates. The master production schedule is not a sales forecast that represents a statement of demand. The master production schedule must take into account the forecast, the production plan, and other important considerations such as backlog, availability of material, availability of capacity, and management policies and goals”. In our modern age of high competition and technology enhancements, this balancing mechanism is a vital tool for management at various levels. At the executive team level, sales and operations planning has become the integrator of all top-level plans. At middle management levels, and on plant floor, MPS spells out in details what need to be produced so that the company can ensure that capacity will be available, that materials will be on hand when needed, and that customer requirements will be satisfied on dates specified by the customers. Master Scheduling thus acts as a buffer between the Sales Forecast and Production as shown in figure 2.1.

MPS is mostly used to plan for independent demand items. Independent demand items are planned on the basis of a forecast. End products, service parts, and spare parts such as, refrigerators, chairs, cupboards and their service and spare parts, are examples of independent demand items.

2.2.2 Make To Stock (MTS)

“A production environment where products can be and usually are finished before receipt of a customers order. Customer orders are usually filled from existing stocks, and production orders are used to replenish those stocks” [2]. “The MTS company produces in batches, carrying finished goods inventories for most, if not all, of its end items” [57]. These are

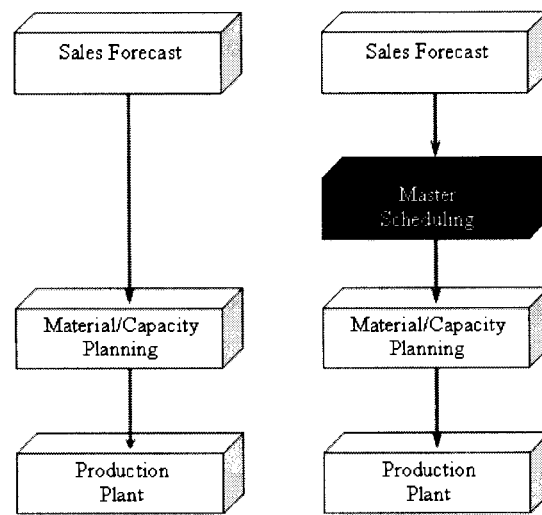


Figure 2.1: MPS in Production [51], by permission

typical off the shelf items we buy from stores, ranging from cutlery, crockery, appliances etc. These companies have their end products planned by MPS. MTS is comparable to Commercial Off The Shelf (COTS) software products like word processors, spreadsheets etc.

2.2.3 Assemble To Order (ATO)

“An environment where a product or service can be assembled after receipt of a customers order. The key components (bulk, semifinished, intermediate, subassembly, fabricated, purchased, packaging, etc.) used in the assembly or finishing process are planned and possibly stocked in anticipation of a customer order. Receipt of an order initiates assembly of customized product. The strategy is useful where a large number of end products (based on selection of options and accessories) can be assembled from common components” [2]. “The ATO firm is typified by an almost limitless number of possible end item configurations, all made from combinations of basic components and subassemblies” [57]. The typical example will be an automobile company, which has unlimited end product combinations. These companies have their components and sub-assemblies planned by MPS.

2.2.4 Make To Order (MTO)

“A production environment where a product or service can be made after receipt of a customers order. The final product is usually a combination of standard items and items custom designed to meet the special needs of the customer. Where options or accessories are stocked before customer orders arrive, the term assemble-to-order is frequently used” [2]. “The MTO company, in general carries no finished goods inventory and builds each customer order as needed” [57]. The example for these kind of production are capital goods manufacturers, like hydrocrackers, refinery equipment et al. These are mostly tailor made products. These companies have their raw materials planned using MPS, as the only thing that can be common between any two projects can be raw material.

This can be compared with special purpose software needed for an organization, where the organization has the basic design ready, and the developers need to only develop the source code.

2.2.5 Engineer To Order (ETO)

“Products whose customer specifications require unique engineering design or significant customization. Each customer order results in a unique set of part numbers, bills of material, and routings” [2]. This kind of production environment is very similar to MTO environment, except that the product is started from the design stage without even having any raw material needed in the stock. These kinds of production houses cater for very exotic products for which even the basic design and raw materials are different from project to project. These are mostly turn-key projects.

In software terminology this would qualify for customer providing broader software requirement guidelines and the developers developing the detailed design, workflow layout, execution plan et al required to develop the software.

2.2.6 Customer Order Decoupling Point

“The Decoupling point is defined as the point that indicates how deeply the customer order penetrates into the flow of goods (Hoekstra and Romme, 1992). The location of the decoupling point within a production process determines, on the one hand, the balance between procurement, production, storage and distribution; on the other hand, it indicates the level of customization and customer service that can be offered. Basically, this means that the customer order decoupling point separates forecast-driven activities, e.g. producing a general product to stock, from customer-driven activities, e.g. producing a specific product to customer order” [58]. Customer Order Decoupling Point (CODP) diagram shown at figure 2.2 gives a glimpse of how various manufacturing environments work.

The figure 2.2 shows the penetration of Customer Order into flow of goods. In MTS scenario, the customer order appears after the flow of goods is complete. In ATO, it comes somewhere in between the production start and completion of finished goods. For MTO and ETO, the production process begins after the company receives an order from the customer.

2.2.7 Manufacturing Environments and MPS

The figure 2.3 best explains the relation between various Manufacturing Environments and MPS.

MTS: In MTS, MPS is used to plan for Finished products. These are off the shelf items. The Customer Order is for the finished goods and there is no lead time for delivery of goods

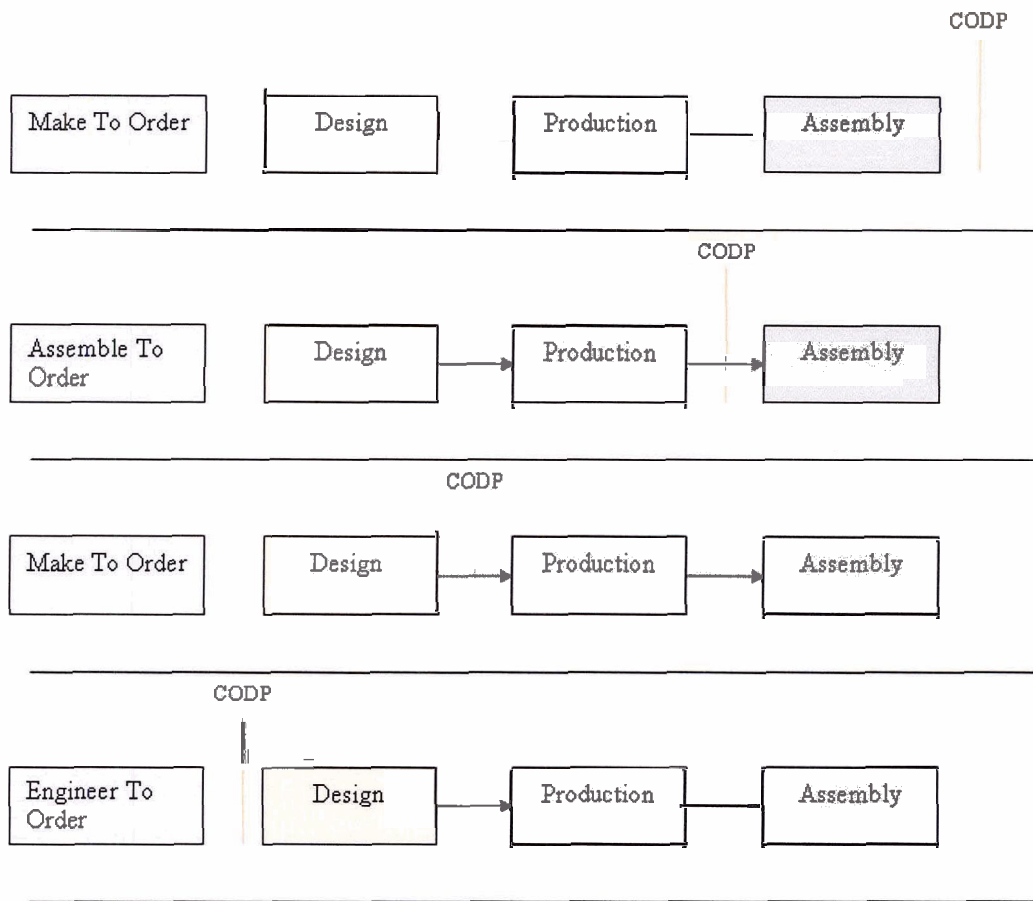


Figure 2.2: Customer Order Decoupling Point (CODP)

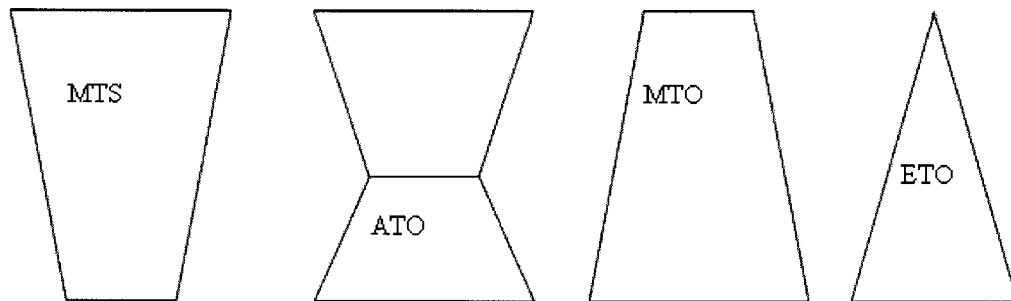


Figure 2.3: Manufacturing Environments

to customer. In MTS, a fix number of finished products are produced.

ATO: In ATO environment, MPS is used to plan for sub-assemblies or parts of final products. The sub-assemblies are parts are assembled after the customer order is received. Unlike MTS, there is a lead time between the customer order and the goods delivery, however, this lead time is generally small. In ATO, fix number of subassemblies and parts are used to plan for a big number of finished products.

MTO: In MTO environment, MPS is used to plan for raw materials. In these kind of scenario, the production house makes specialized products and does not start production till receiving customer order. These products must cater to a specific industry or should be of similar kind. Unlike MTS and ATO, MTO has much bigger delivery lead time.

ETO: ETO environment caters to very specialized industry, where they are never sure of any repeat orders. Hence, no stock is maintained and the production house starts designing the product, after receiving the order. The raw materials are then ordered. ETO has the longest delivery lead time as event the raw material has to be ordered. There are no stocking levels for ETO environment.

2.2.8 Detailed Analysis of Assemble To Order Environment

As per Fogarty et al [28], a firm operates in an assemble-to-order mode whenever the time a customer is willing to wait for delivery is greater than the time required to assemble the product from components and is less than the time required to fabricate the components and assemble the product. Most assemble-to-order firms have an hourglass bill of material. An

hourglass bill of material is a complex assembly, made from a few major subassemblies that in turn are made from a number of components. The hourglass bill of material usually arises from the number of different ways one can combine modules to produce an end item. There are virtually an infinite number of ways to configure a car from a small number of possible chassis, body style, body color, engine, transmission, and interior option configurations. A customer usually is willing to wait a small amount of time provided he or she can specify precisely the combination to be used in creating the end item. Because almost any item can be assembled in a short time, assembly lead time is short for almost all products. However, manufacturing lead time is usually quite long. It is therefore not surprising to find that most companies having hourglass-shaped bills of material operate assemble-to-order, and vice versa. Let's evaluate a simple bill of material for a computer manufacturer. Let's suppose that he offers following choices to his customers

- 3 Motherboards
- 4 Video Cards
- 3 Sound Cards
- 6 Keyboards
- 8 Mice

With these limited choices offered by the supplier, we can calculate that the customer is given an option to pick one of 1728 ($=3 \times 4 \times 3 \times 6 \times 8$) choices. If the supplier was to use MTS scenario, even if the supplier has to stock 100 units of each kind, it would amount to 172,800 computers and at the cost of \$1000 a computer, the inventory would be whopping \$172,800,000. The Inventory Turnover Ratio, defined as APICS [2] is the ratio of annual sales turnover of an organization to the average inventory held by the organization during the course of a year. For same sales, higher the average inventory, lower is inventory turnover ratio and vice versa. A higher inventory means higher carrying costs, higher obsolescence rate and higher value of inventory damaged due to storage. All these expenses are taken out directly from the profitability of an organization. Thus, an organization with higher turnover ratio is more efficient and profitable than the one with lower turnover ratio. As per Comerford [14], "Low inventory turnover can indicate poor liquidity, possible overstocking, obsolescence, or, in contrast to these negative interpretations, a planned inventory

buildup in preparation for future material shortages”. On the contrary, if the supplier uses MTO scenario, no customer would be willing to wait for as long as a month to deliver a simple desktop computer. The only viable option for the supplier is to choose ATO and run MPS for the parts/subassemblies. The supplier can then decide to stock a few hundred units of each kind of parts and on the basis of the order/demand, can assemble the computer and deliver them. At the same time, if a particular part is out of stock, the supplier has an option to substitute with a higher quality part or with a discount and thereby saving the order or a valuable client.

2.2.9 Advantages of ATO

“In the “faster, better, and cheaper” information age, rapid technology innovations have significantly increased the pace of product obsolescence. In particular, for an assembled product, its life cycle is directly determined by the life cycles of its constituent components; that is, technology obsolescence at the component level affects the system at the product level” [14]. Thus ATO plays an important role where it is easier to get rid of an obsolete component, rather than the entire finished product containing the component.

The higher the Inventory Turnover Ratio is, the less are the inventory carrying costs, and hence the better profits for an organization. Inventory Turnover Ratio is defined as the ratio of Cost of goods sold and average or current period inventory [33]. The lesser is inventory, higher is the turnover.

ATO helps in providing better options to customers. ATO provides the supplier with an option to replace a part with a better substitute, thereby saving an order and importantly, retaining the customer.

2.3 Software Reusability

Reusability is an important aspect of software design. One of the ways to increase software productivity is to reuse existing code, designs and specifications [62]. According to Judith Barnard [5], there are number of factors that decide the reusability of any software. They range from low module complexity, good documentation, few external dependencies, proven module reliability on general term to better encapsulation, cohesiveness, less complexity, and so on. Software reusability enhances both productivity as well as the quality of the software.

- Improved productivity

Since the software is reused, there are minimal development efforts associated for subsequent uses and hence the productivity improves. However, it is only beneficial if the software is reused. Frakes et al [29] discussed about various metrics used to measure the cost reduction in reusing the software and analyze that the development efforts in developing a reusable software is productive only when the software is reused, otherwise the cost and efforts involved in making a software reusable are lost.

- Improved Reliability/Quality

Software reusability not only saves the development time for the software, but also the testing and scalability issues because the software is already tested for its functional efficiency and in all probabilities tested for scalability as well. According to Voas et al [56] software reuse can be more readily enabled if the testing of the code in the previous environment is still applicable to the new environment. Since the code has passed through several implementations, it is already survived the various requirements, which might not have been feasible otherwise. “It is commonly accepted that the easiest software bugs to find are discovered most quickly. It requires increasingly more effort to find and correct successive bugs, so that eventually bugs become relatively expensive to track down” [42].

2.3.1 Software Reusability and LMS

Much emphasis has been laid on the reuse of content on various LMS and the issue of content interoperability is supported through open source standards like SCORM [38], IMS [15] and Cancore [10]. While the content portability is supported by these robust standards, there is not much effort in suggesting the reuse of software or applications across various LMS. One of the major efforts in this direction is the Open Knowledge Initiative OKI [48] by MIT.

Open Knowledge Initiative (OKI)

“The Open Knowledge Initiative develops specifications that describe how the components of an educational software environment communicate with each other and with other enterprise systems” [48]. The OKI team provides architectural and technical leadership for software initiatives in higher education to ensure that the next generation of eLearning software will integrate seamlessly with the educational enterprise.

The deliverables as specified by the OKI white paper [13] are

- A set of API, described abstractly and written in Java™, that enable educational technology to provide and access commonly used services. These API are freely available, and are supported by reference examples of working learning platforms and documentation of the architectural assumptions underlying the API.
- Components of learning management software developed by MIT (Stellar), by Stanford University (Course Work), and by the University of Michigan (Chef). This code will be made available as open source code that can be freely incorporated into learning system components built by universities or by commercial software vendors.

The white paper was written in September 2002. Since then, MIT, Stanford University, University of Michigan, Indian University and uPortal consortium have come together to develop SAKAI, based on the OKI architecture.

OKI Architecture

According to Scott et al [55], the OKI architecture is described by two service layers viz; Common Services and Educational Services. Common services provide institutional infrastructure and other fundamental services, basically acting as a backbone for higher level services and applications. The Educational layer provides framework for the developers to develop any application suited for their requirements. The OKI architecture is shown in figure 2.4

As shown in the figure 2.4, Common Services reside at the bottom and interact with institutional infrastructure, such as the underlying database, security system et al. Common Services cater to networking and to authentication support, and are basic necessity for most systems, both educational or commercial eCommerce systems. They are more for use by the systems support group rather than the eLearning developers. On top of the Common Services, sit Educational Services, which are very specific to educational needs. Educational Services cater to basic needs of an educational institution such as Course Management, Assignment Grading, Team Management, and so on. Any new application added to the LMS developed on OKI architecture is expected to extend the Educational Services so that the application can be shared based on the OKI architecture. OKI group has developed an

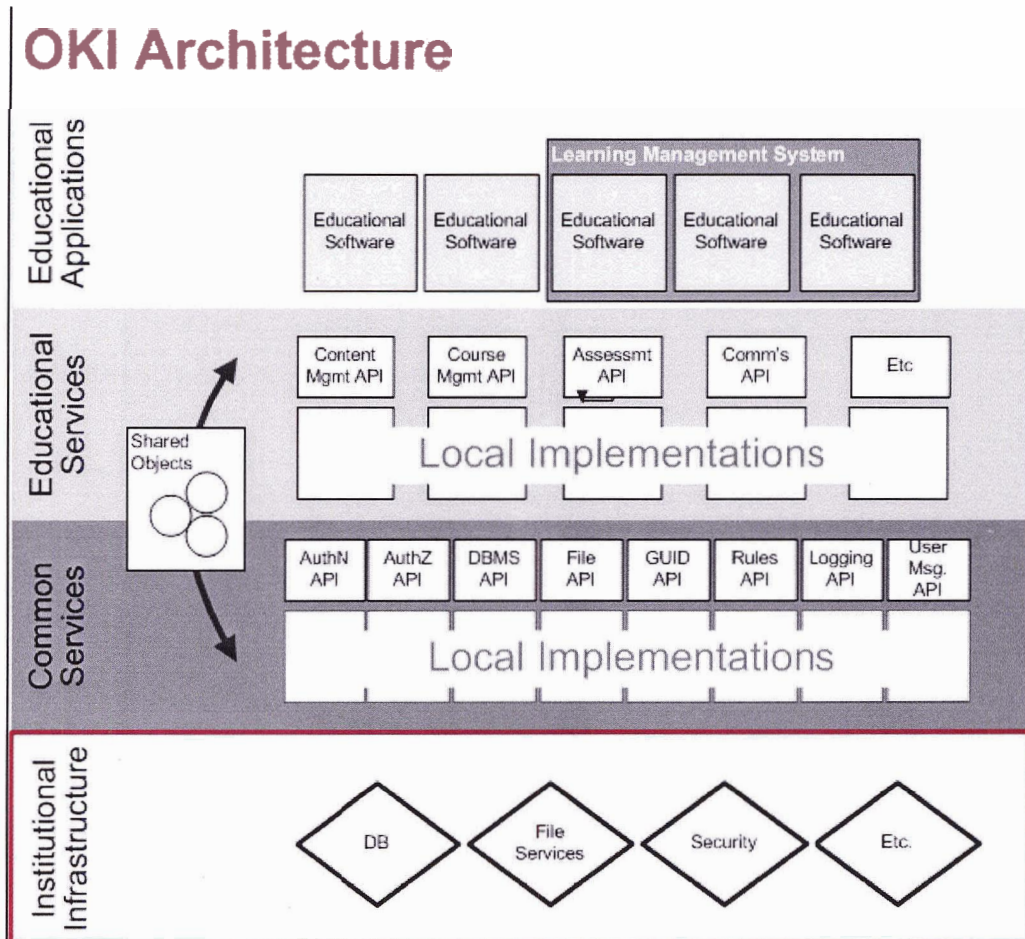


Figure 2.4: OKI Architecture [55], by permission

API of interfaces called OSID [25] to guide the development of new services or to extend the existing services.

2.3.2 OSID(OKI Services Interface Definition)

“The Open Knowledge Initiative has produced a series of Open Service Interface Definitions (OSID) informed by a broad architectural view of the educational technology landscape” [25]. Both the Common Services and the Educational Services are developed on the OSID framework.

According to Kahn [35], OSID place a boundary between the service providers and the service consumers. Service Providers develop services that cater to some specific requirements and Service Consumers use these services while developing their applications. For example, a service provider provides the service to check user authentication and then award authorization to access a particular course. Instead of rewriting the authentication/authorization for every application, the developer (Service Consumer) for a new application can make use of the existing Authentication/Authorization service.

The OKI/OSID team have developed some of the OSID services like Agent, Assessment, Authentication, Course Management and so on. Most of the OSID are in the early stages of development and as per the OKI/OSID team they are not yet fully matured.

Not many LMS are based on the OKI/OSID framework mostly because the project is not matured yet and secondly because even though the project is very novel concept, it has steep learning curve. SAKAI is one of the major LMS initiatives that uses OKI/OSID.

2.3.3 SAKAI

“The SAKAI Project is a community source software development effort to design, build and deploy a new Collaboration and Learning Environment (CLE) for higher education. The Project began in January, 2004” [23]. SAKAI is one of the most popular LMS projects around the world, with more than 80 members, mostly educational institutions, around the world. It is a collaborative project by the University of Michigan, Indiana University, Stanford University, MIT and uPortal consortium.

SAKAI is based on the OKI/OSID framework, however they have extended the existing OKI/OSID framework to suit their requirements. SAKAI is built on top of University of Michigan’s CHEF (CompreHensive Collaborative Framework) and promotes the extensive

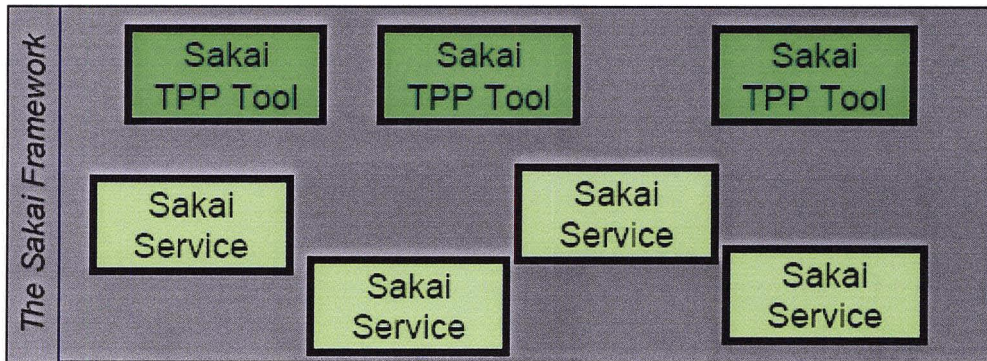


Figure 2.5: SAKAI Framework [17], by permission

use of open source or free technologies like Java Server Pages (JSP), JavaTM Servlets, Spring, Hibernate, Java Server Faces (JSF), Maven et al. SAKAI team plans to reduce the dependency on the CHEF legacy tools with each release till eventually the whole system is built on the new architecture. The architecture discussed here belongs to version 1.5 of SAKAI.

The SAKAI Abstract Architecture is built on a JavaTM Framework supported with extensions of OKI/OSID. The tools or applications layers are built on the abstract architecture. The applications are developed using SAKAI Technology Portability Profile (TPP). This section describes specific technologies chosen for the layers identified in the SAKAI Architecture that leads to the SAKAI Tool Portability Profile. The Tool Portability Profile (TPP) provides detailed guidance with respect to the development of tools that will operate within the SAKAI JavaTM Framework as shown in figure 2.5.

“In addition to supporting SAKAI TPP tools, the framework also supports a number of other capabilities such as portal integration and non-TPP application integration” [17].

SAKAI is an excellent approach towards developing a matured open source LMS (or Collaborative Learning Environment (CLE)) backed by premier institutions and an excellent team of developers. It is an ideal solution for achieving interoperability across multiple LMS running across multiple institutions. However, the project is not expected to reach its maturity till mid 2007, at the least. Besides, SAKAI technologies and architecture have a steep learning curve which makes it less interesting to the developers outside the core group.

A good support is necessary for supporting and helping the new developers to be able to contribute to the project, which unfortunately is missing. One needs to wait for a while before being able to convincingly comment about the success of SAKAI.

2.4 Open Source/Free Technologies

There are many open source or free to use web development technologies available today. Some of the most prominent ones are eXtensible Markup Language (XML), Java™ and Java™ related technologies like Servlets, Java™ Server Pages (JSP), JAXP (Java™ API for XML Parsing), eXtensible Stylesheet Language Transformation (XSLT), PHP, Python, Zope et al. In this thesis, the emphasis is laid on technologies that are relevant to this thesis. These are XML, XSLT, Java™ Servlets and JAXP.

2.4.1 eXtensible Markup Language (XML)

eXtensible Markup Language (XML) is a markup language and is a World Wide Web Consortium (W3C) [16] recommendation. “Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (ISO 8879)” [60]. XML is very similar to HTML and contains HTML like tags. However, XML differs in a way that it is more strict about the tagging, like every tag must have a closing tag and unlike HTML it provides the user with a flexibility of creating their own tags. Thus, XML is considered as a metalanguage, a language used to describe another language(s), since it allows the users to define their own tags [3].

XML document is a text file, and that makes it platform independent . XML is very powerful in exchanging information and is the basis of many standards of exchanging information over the internet. There is a spate of technologies using XML for exchanging information and the growing number of commercial as well as open source parsers for parsing XML documents. XML document has an optional DTD (Data Type Definition) or XML Schema. The DTD or Schema contains rules about the data and the order in which it should appear in any XML document. An XML document is an instance of its schema or DTD. Though a DTD or a schema is not mandatory, it is always advisable to have one. It is always prudent to develop applications based on Schemas rather than an XML document, which is just a particular instance of a schema or a DTD.

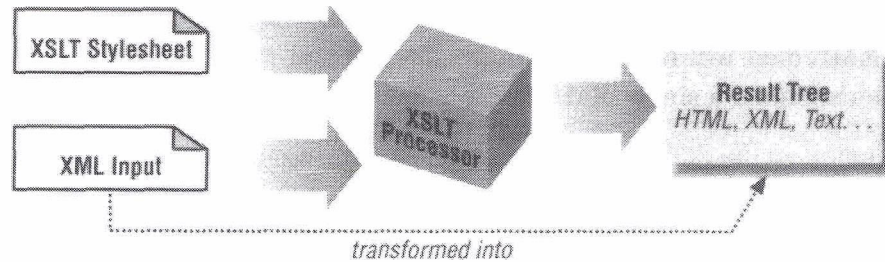


Figure 2.6: XSLT Transformation [9], by permission

2.4.2 eXtensible Stylesheet Language Transformation (XSLT)

eXtensible Stylesheet Language Transformation (XSLT) is a stylesheet language and like XML is also a W3C recommendation. XSLT “is a language for transforming XML documents into other XML documents” [61]. The XSLT document is also a well formed XML document.

The purpose of an XSLT document is to transform an XML document into other tree based document, which can be an HTML, XML, WML(Wireless Markup Language) or any other form of tree based document. This is achieved by feeding an XML document and XSLT document into an XSLT processor. The processor based on the information contained in these documents, transforms the XML file into the desired output. The figure 2.6 shows how the transformation takes place. “While originally designed to serve as a stylesheet, to map XML into HTML, XSLT is increasingly used in other applications, such as querying and transforming XML data” [34]. XSLT gives the flexibility to separate the presentation layer from the business logic.

Both the major browsers (Internet Explorer from Microsoft and Firefox from Mozilla) have built in support for XSLT.

2.4.3 Java™ API for XML Processing (JAXP)

“The Java™ API for XML Processing (JAXP) enables applications to parse validate and transform XML documents” [24]. JAXP also has complete support for XML 1.1, Document Object Model (DOM) L3, XInclude, and Simple API for XML (SAX) , XPath, XSLT

standards. JAXP offers typical JavaTM portability by making the whole API independent of any particular XML parser or XSLT processor. When using JAXP, the code can avoid dependencies on specific vendor tools, allowing flexibility to upgrade to newer tools when they become available [9].

We thus observed that ATO is a useful concept in Production Management for reducing supply lead times while catering to customer needs. The ATO can be extended to LMS, wherever the source code is available. This can be achieved by modifying the typical request response using XML and XSLT and other open source technologies. The reduction in the development efforts needed to re-deploy an application helps institutions to share the success of an application with institutions using different LMS.

Chapter 3

METHOD

Through this thesis, we propose the means to develop Assemble-To-Order LMS. We propose a design that makes most of the code portable and eliminates proprietary database dependency. The typical request response system used on the Internet is modified on server side to reduce the database dependency by inserting an XML/XSLT layer between the backend database and the request response system. JAXP is used to both convert the data retrieved from the database into XML format and convert the XML data into database format. The detailed LOC matrix and FP analysis is carried for the Event Sign-up Tool and FP analysis is carried out on three applications selected from EduTools [26] site.

3.1 Typical Request Response System

A typical request response system on the Internet works as follows

- A request is sent to the webserver.
- Webserver directs the request to a JavaTM Servlet.
- The servlet queries the backend database, if needed, for retrieving information or for updating the database, and generates a response for the client.

The system is shown in figure 3.1.

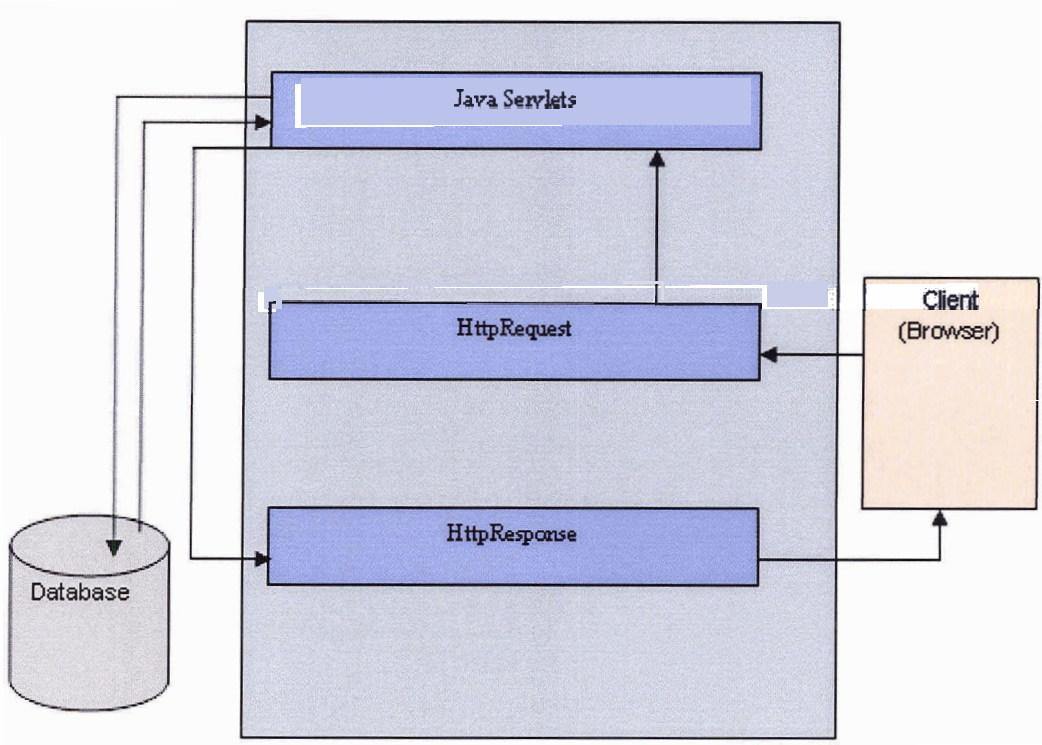


Figure 3.1: Typical Request Response System

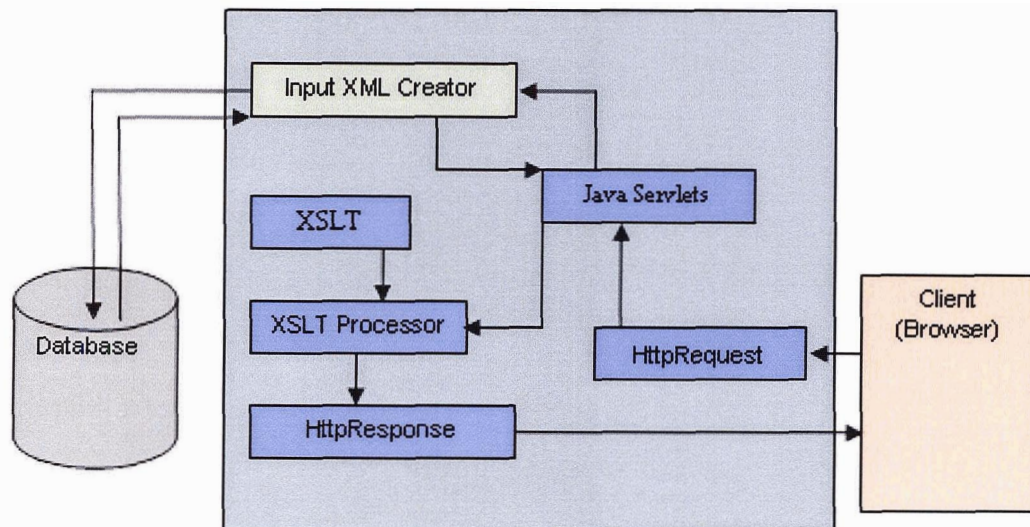


Figure 3.2: ATO Query System

3.2 ATO Request Response System

The following changes are made to the existing request response system for it to become an ATO Request Response System. Unlike a typical request response system, the request is handled differently for retrieving information (querying) and for updating the database.

- Querying:
 - A request is sent to the webservice.
 - Webserver directs the request to a JavaTM Servlet.
 - Instead of querying the database directly, the servlet requests for an XML input from Input XML Creator (IXC).
 - IXC queries the database and sends back the information to JavaTM Servlets in XML format.
 - JavaTM Servlet then renders the XML using XSLT and sends it back to the client

The system is shown in figure 3.2.

- Updating:
 - A request is sent to the webserver.
 - Webserver directs the request to a Java™ Servlet.
 - Instead of updating the database directly, the servlet generates a streaming XML output using Output XML Creator (OXC).
 - OXC is then fed to the Output XML Processor which in turns updates the database.
 - OXC also sends error or success message back to servlets.
 - Depending upon the message it got, the servlets then displays the success or error page. For success page, it may again need to query data from the database.

The system is shown in figure 3.3.

3.2.1 The Design for ATO LMS

The application is designed to take its input from XML streams rather than from the proprietary database. The result of a business process is converted into an XML stream rather than using it to update the database. Any institution implementing such an application would need to develop its own varying components to convert the input/output XML streams into their own database schemas compatible information. The design can be broken into two parts viz;

- 1) Fixed Part
- 2) Variable Part

3.2.2 Fixed Part

As the name suggests, this part of the application will remain unchanged and does not need any modification by the implementing institution. This consists of the following layers

- **Presentation Layer:** This layer is responsible for designing the user interface for the end user. The layer is implemented in XSLT to separate the business layer from the presentation layer. The institution has an option to modify the user interface depending upon their needs as long as all the parameters needed for the business

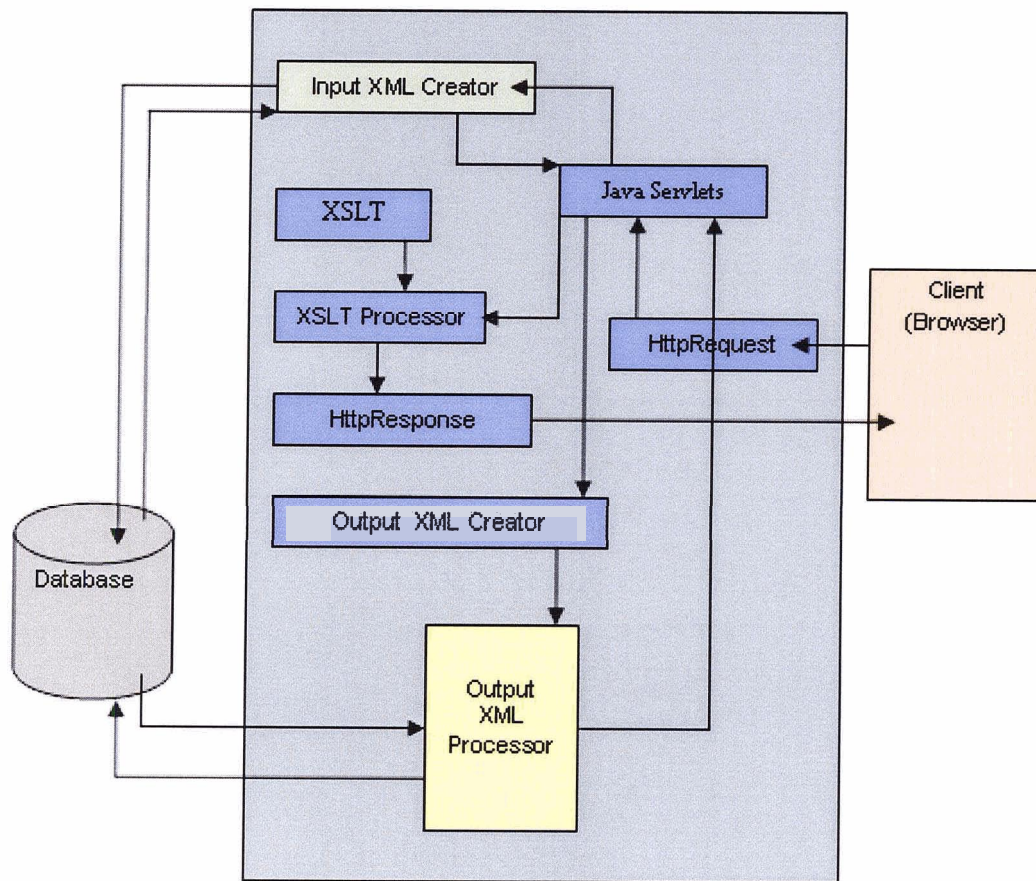


Figure 3.3: ATO Update System

layer are agreed upon. Specific XML schemas are provided to enable organizations to implement their own user interface.

- **Business Layer:** This layer is responsible for all the business logic, exception handling, and generating streaming XML documents needed to update the database. The XML document is then forwarded to the variable part which in turn updates the proprietary database. This layer also interacts with the variable part to fetch the information needed to display the content. XML schemas are provided for every XML document generated or expected.
- **XML Layer:** As against a database layer in a typical web application, this approach uses XML documents for retrieving or updating information. The XML documents interact with proprietary database for accessing or updating the information.

3.2.3 Variable Part

Every institution has its proprietary database. This makes it practically impossible to develop a Make-To-Stock application. Hence a varying part needs to be developed in order for the ATO application to work. Here, the code parses an XML input stream and depending upon the need either retrieves information from the backend database or updates the database. The XML schemas are provided as guidelines for developing the varying part.

3.3 Evaluation

The evaluation is carried out in two steps. In first step, a sample application called Event Sign-up Tool (ESUT) is developed for the purpose of demonstrating the design concept. The application has two parts,

- Instructor (or TA) part
- Student part

For the purpose of this thesis, only the instructor part is developed. The ESUT application is developed for a Course Management System developed in-house, named CMS, by Simon Fraser University. The variable part of the application was modified to make it portable to another open source LMS called Eledge [47], developed by the University of

Create New Event

Field trip to vanc art gallery	Event Name
Since the class size is big, we are planning it on three days. Give your two best choices that work	Event Description
3	No. of Sessions
2	No. of Choices
2006-12-31	Signup End Date (yyyy-mm-dd)
Create	

Figure 3.4: Create Event Page

Utah. CMS uses MS SQL Server as the backend and Eledge uses MySQL as The backend database.

In second step, three applications from EduTools [26] are evaluated using FP analysis.

3.3.1 Event Sign-up Tool (ESUT)

Event Sign-up Tool is an application designed to help faculty or TA organize an event, be it a field trip, class schedule or any other type of event. An event has a number of options which are called sessions. Students are allowed to pick one or more sessions. There is a deadline for each event, after which students are not allowed to register. The faculty has to go to the “Create New Event” page to create an event. The page is shown in figure 3.4

The page contains information about the event. The fields are explained below:

- **Event Name:** This field captures the name or the title of the event.
- **Event Description:** This field presents a description about the event.

Create New Session (1 of 3)

14th Jan	Session Name
We meet at 9AM in the lobby of the gallery.	Session Description
35	No. of Seats
Create	

Figure 3.5: Create Session Page

- **No. Of Sessions:** This field contains information about how many options the students get. For example: As shown in the example above, a Vancouver Art Gallery visit might be done on three days, in which case the value for this field becomes 3.
- **No. Of Choices:** This field contains information about the choices given by the students. For example: as in example above, a student is invited to give two of his or her best choices. It might be because the students actually need to go only on two days, or to provide flexibility to the instructor to allocate one or more students to their second choices, depending upon the demand for a particular day.
- **Sign-up End Date:** This field is to capture the registration deadline for students. After this date, students are not allowed to register for the event.

Once the event is created by clicking on “Create” button, the instructor is shown the interface in figure 3.5.

Since there are supposed to be 3 sessions in the event, and the first screen is “Create

New Session (1 of 3)". The fields of the page are explained below.

- Session Name: The name of title of the session.
- Session Description: The detailed description about the session.
- No. Of Seats: Maximum number of enrollments allowed in a session.

Once first session is created, the instructor is shown the interface/screen to "Create New Session (2 of 3)" page and then further to "Create New Session (3 of 3)" page. After that, the instructor is shown the "ESUT Main Page" as shown in figure 3.6



Figure 3.6: ESUT Main Page

The created event is thus shown on the main page. At this stage, the event is not posted. By clicking on post button, the event is posted to students for registering. Once the event is posted, the “Post Event” button is disabled as is evident in the “My Class Schedule” event. There is another button for deleting an event. In a live application, there are other features expected of this application. However, for the purpose of this thesis, only these two features are developed and demonstrated.

3.4 ATO Source Codes

The source codes are available at appendix B . The source code consists of three implementations viz;

- **ATO-ESUT-basecode:** This is the code that will be available for organizations to download. This is base code, providing fixed part of the code and the framework for developing the variable part. The instructions for the implementing organizations on how to use this source code in their organizations should be provided. These mostly would consist of changing the variable part of the code. An Ant [1] base build.xml file is provided alongwith the source code to help the end-user in compiling the code. The build file is shown in appendix C.
- **CMS-ESUT:** This is the source code used for implementing ATO-ESUT-basecode into SFU Course Management System.
- **Eledge-ESUT:** This is the source code used for implementing ATO-ESUT-basecode into Eledge Learning Management System.

3.4.1 ATO-ESUT-basecode

Following are the components of this code.

- **Package esutfixed:** This contains the business logic and should not be edited. The full name for the package is ca.sfu.elinc.esutfixed.
- **Package esutvariable:** This part is responsible for interacting with the backend database and needs to be coded by the implementing organization. This code is not implemented at this time. The guidelines and framework to develop this code are provided in the framework. The full name for the package is ca.sfu.elinc.esutvariable.

- XSLT Documents: These parts contain the Graphical User Interface (GUI) needed for the application. It is upto the institution to modify it or not depending on their requirements.
- Package javaxslt: This is represented by ca.sfu.elinc.javaxslt package and contains classes for helping in generating XML documents and processing them with XSLT to produce the HTML output.
- XML Schemas: The XML-schemas act as guidelines for creating or processing XML documents. The list of schemas used are included in appendix A.

3.4.2 Package esutfixed

This package has following components

- Package util: This package contains only one utility class EsutUtilities.java, which is used for providing miscellaneous utilities for the application.
- EsutDeleteEvent.java : This is a servlet that is used for deleting an event.
- EsutMain.java: This is the servlet that is used for rendering “ESUT Main Page”
- EsutNewEvent.java: This is a servlet that is used for managing “Create New Event” page and converting event values into XML format.
- EsutNewSession.java: This is a servlet that is used for managing “Create New Session (X of Y)” page and converting session values into XML format.
- EsutPostEvent.java: This is a servlet that is used for posting an event.
- EsutXMLInputCreatorNewSession.java: This class is used for creating XML inputs for a new session.
- EsutXMLOutputCreatorDeleteEvent.java: This class is used for creating the XML needed to delete an event.
- EsutXMLOutputCreatorNewEvent.java: This class is used for creating XML document with event values on receiving inputs from an end-user. This class is used by EsutNewEvent servlet to get the input XML for passing it on to EsutProcessOutputXML class in esutvariable package.

- `EsutXMLOutputCreatorNewSession.java`: This class is used for creating XML document with session values on receiving inputs from an end-user. This class is used by `EsutNewSession` servlet to get the input XML for passing it on to `EsutProcessOutputXML` class.
- `EsutXMLOutputCreatorPostEvent.java`: This class is used for creating the XML needed to posting an event.

3.4.3 Package `esutvariable`

This package will vary from one organization to another and hence need to be implemented. The classes are populated with empty methods which need to be implemented. The package has following components.

- `EsutProcessOutputXML.java`: This class is used for processing all the output XML for updating the database. This is used for saving new events, delete existing events, posting existing events and saving new sessions. The following methods are implemented in this class
 - `processOutputXMLNewEvent (ByteArrayOutputStream, PrintWriter)`: This method is used for saving new event information provided by the XML document contained in `ByteArrayOutputStream` parameter. If the event is created successfully, the method should return `eventId` or else should return `-1`. The XML provided is based on `SaveEvent.xsd` schema.
 - `processOutputXMLNewSession(ByteArrayOutputStream, PrintWriter)`: This method is used for saving new session information provided by the XML document contained in `ByteArrayOutputStream` parameter. If the session is created successfully, the method should return `boolean true` or else should return `boolean false`. The XML provided is based on `SaveSession.xsd` schema.
 - `processOutputXMLDeleteEvent(ByteArrayOutputStream, PrintWriter)`: This method is used for deleting an event based on the information provided by the XML document contained in `ByteArrayOutputStream` parameter. If the event is deleted successfully, the method should return `boolean true` or else should return `boolean false`. The XML provided is based on `DeleteEvent.xsd` schema.

- `processOutputXMLPostEvent(ByteArrayOutputStream, PrintWriter)`: This method is used for posting an event based on the information provided by the XML document contained in `ByteArrayOutputStream` parameter. If the event is posted successfully, the method should return boolean `true` or else should return boolean `false`. The XML provided is based on `PostEvent.xsd` schema.
- `EsutXMLInputCreatorMain.java`: This class is used for creating an input XML for the main page. The class has only one method.
 - `getInputXML(HttpServletRequest)`: This method should use the `HttpServletRequest` object to return a list of events in XML format. The XML should be returned as an instance of `ByteArrayOutputStream` object and should be based on `ListEvents.xsd` schema.
- `EsutXMLInputCreatorNewEvent.java`: This class is used for creating an input XML for the “Create New Event” page. The class has only one method.
 - `getInputXML(HttpServletRequest)`: This method should use the `HttpServletRequest` object to return the XML document needed to initialize the create event page. The XML should be returned as an instance of `ByteArrayOutputStream` object and should be based on `NewEvent.xsd` schema.

3.4.4 XSLT Documents

The use of XSLT documents give flexibility to the developers to change the user interface according to their requirements. The following XSLT documents are used in the application.

- `createEvent.xslt`: This XSLT file is used for displaying the “Create New Event” page. The XML used by this document is based on `NewEvent.xsd` schema.
- `createSession.xslt`: This XSLT file is used for displaying the “Create New Session (X of Y)” page. The XML used by this document is based on `NewSession.xsd` schema.
- `listEvents.xslt`: This XSLT file is used for displaying the “ESUT Main” page. The XML used by this document is based on `ListEvents.xsd` schema.
- `reEnterEventData.xslt`: This XSLT file is used for redisplaying the “Create New Event” page if the incorrect inputs are fed. A sample for this page is shown in

figure 3.7. The XML used by this document is based on ReEnterEventData.xsd schema. There are some more error messages which are not shown in the figure.

- reEnterSessionData.xslt: This XSLT file is used for redisplaying the “Session X of Y” page if the incorrect inputs are fed. A sample for this page is shown in figure 3.8. The XML used by this document is based on ReEnterEventData.xsd schema. There are some more error messages which are not shown in the figure.

3.4.5 Package javaxslt

This package has multiple classes that assist in generating XML documents and also in processing XML and XSLT documents to generate the required output, which is mostly HTML. The code in this package can be used by any organization in getting the necessary XML documents needed to implement the esutvariable package. The package contains following components.

- AbstractXMLReader.java: This class is an implementation of SAX2 XMLReader interface. The intent of this class is to make it easy for subclasses to act as SAX2 XMLReader implementations. This makes it possible, for example, for them to emit SAX2 events that can be fed into an XSLT processor for transformation [9].
- XMLProcessor.java: This class is used for converting an XML document into another document with help of an XSLT file. There are two methods in this class. viz;
 - changeFormat(PrintWriter, ByteArrayOutputStream, File): The method provides XML in ByteArrayOutputStream and XSLT is supported by a File object. It returns a ByteArrayOutputStream of resultant document. PrintWriter object provides optional feature of displaying the process to the end-user.
 - changeFormat(PrintWriter, ByteArrayOutputStream, String): This method is an overloaded method for changeFormat mentioned above. Here the file name is supplied as a String.
- XMLTools.java: This class is used for providing XML tools. There is only one method in this class viz;

Create New Event

- Name can not be blank
- Description can not be blank
- Number of Sessions can not be blank
- Maximum choices can not be blank
- End date can not be blank

The form consists of several input fields and a button, arranged vertically. From top to bottom: a single-line text input field for 'Event Name'; a large multi-line text area for 'Event Description' with a small 'x' icon in the bottom right corner; a single-line text input field for 'No. of Sessions'; another single-line text input field for 'No. of Choices'; a single-line text input field for 'Signup End Date (yyyy-mm-dd)'; and finally, a rectangular button labeled 'Create'.

Figure 3.7: Re-enter Event Data

Create New Session (1 of 3)

- Name can not be blank
- Description can not be blank
- Number of Seats can not be blank

The form consists of three vertically stacked input fields and a 'Create' button at the bottom left. The first field is labeled 'Event Name', the second is labeled 'Event Description', and the third is labeled 'No. of Sessions'. Each input field has a small square icon in its bottom right corner, likely representing a cursor or a focus indicator. The 'Create' button is a simple rectangular box with the text 'Create' inside.

Figure 3.8: Re-enter Session Data

- `populateContentHandler(ContentHandler, String, Hashtable)`: This method populates the interface `org.xml.sax.ContentHandler`. If the input `Hashtable` contains object `String` as value, then the key becomes the tag and the string as the value contained in the tag. If the object is a `Hashtable`, then it is sent into the recursive loop. If the object is a `Vector`, then all the elements in the `Vector` should be `Hashtables`. The key in that case becomes the parent tag and the `Hashtables` contained in the `Vector` are sent into the recursive loop.

3.4.6 XML Schemas

The XML schemas used in the application can be categorized into

- **Internal Schemas:** These schemas are used by `esutfixed` package and are not supposed to be used by implementing organization. The following schemas are internal schemas.
 - `NewSession.xsd`: This schema is used for displaying “Create New Session (X of Y)” page. Since these pages are displayed only after event is created, the system automatically gets the information needed to create this schema. The XSLT file used with this schema is `createSession.xslt`
 - `ReEnterEventData.xsd`: This schema is used when user feeds invalid input in “Create New Event”. The system detects those errors and display all the error messages, retaining all the values. So, that the user does not have to re-enter them again.
 - `ReEnterSessionData.xsd`: This schema is similar to `ReEnterEventData.xsd`, except that it is used for catching the incorrect input in “Create New Session (X of Y)” page.
- **External Schemas:** These schemas are used by `esutfixed` package and should to be used by implementing organization as guidelines. This can further be classified into two categories viz;
 - **Schemas for supplying XML:** These schemas are used for supplying XML documents by `esutfixed` package to `esutvariable` package. These XML document are processed by `esutvariable` classes, mostly for updating the backend database.

These schemas are used by `EsutProcessOutputXML.java` class. The schemas falling in this category are

- * `DeleteEvent.xsd`: This schema is used to provide information to `esutvariable` package for deleting an event. The XML supplied is processed by `processOutputXMLDeleteEvent(ByteArrayOutputStream, PrintWriter)` method of `EsutProcessOutputXML.java` class.
 - * `PostEvent.xsd`: This schema is used to provide information to `esutvariable` package for posting an event. The XML supplied is processed by `processOutputXMLPostEvent(ByteArrayOutputStream, PrintWriter)` method of `EsutProcessOutputXML.java` class.
 - * `SaveEvent.xsd`: This schema is used to provide event information to `esutvariable` package for saving it in the database. The XML supplied is processed by `processOutputXMLNewEvent(ByteArrayOutputStream, PrintWriter)` method of `EsutProcessOutputXML.java` class.
 - * `SaveSession.xsd`: This schema is used to provide session information to `esutvariable` package for saving it in the database. The XML supplied is processed by `processOutputXMLNewSession(ByteArrayOutputStream, PrintWriter)` method of `EsutProcessOutputXML.java` class.
- Schemas for generating XML: These schemas are used for generating XML documents by `esutvariable` and which are then pulled by `esutfixed` package. These XML document are processed by `esutfixed` classes, mostly for displaying the GUI to the end-user. These schemas are generated by `EsutXMLInputCreatorMain.java` and `EsutXMLInputCreatorNewEvent.java` classes. The schemas falling in this category are
- * `ListEvents.xsd`: This schema is used to generate XML document containing information about the main page. The `getInputXML(HttpServletRequest)` method of class `EsutXMLInputCreatorMain.java` is used to generate the XML document. The XSLT file used for processing this information is `listEvents.xslt`.
 - * `NewEvent.xsd`: This schema is used to generate XML document containing information about the “Create New Event” page. The `getInputXML(HttpServletRequest)` method of class

EsutXMLInputCreatorNewEvent.java is used to generate the XML document. The XSLT file used for processing this information is createEvent.xslt.

3.4.7 CMS-ESUT

CMS-ESUT is first implementation of ATO-ESUT. It is implemented on the Course Management System(CMS) of Simon Fraser University. CMS is mostly used by SFU Surrey campus students. As required, only esutvariable package is customized to talk to the backend database. CMS uses MS SQL Server as backend database. The detailed source code for this implementation can be checked at appendix B.2. Some portion of the code has been not available there due to sensitive nature of the information there. For the purpose of the thesis, we would show implementation of the method processOutputXMLNewEvent(ByteArrayOutputStream, PrintWriter) of class EsutProcessOutputXML.java for showing how backend database is updated and show implementation of method getInputXML(HttpServletRequest) from class EsutXMLInputCreatorMain to show how the information is queried from the database.

processOutputXMLNewEvent

Please refer code for this method at appendix B.2.2 under class EsutProcessOutputXML.

getInputXML

The class EsutXMLInputCreatorMain is extension of class AbstractXMLReader, that implements SAX2 XMLReader interface. The parse(InputSource) is an abstract method in AbstractXMLReader and is implemented by EsutXMLInputCreatorMain. Please refer to this code at appendix B.2.2 under class EsutXMLInputCreatorMain.

This method uses the method parse(InputSource). The implementation of parse(InputSource) can also be checked at appendix B.2.2 under class EsutXMLInputCreatorMain.

The method parse(InputSource) uses another method called populateContentHandler(ContentHandler, String, Hashtable). This method is developed to populate ContentHandler interface with values from Hashtable parameters. The String parameter acts as root of the XML document thus generated. The method uses following rules

- If Object encountered in `genDetails` is a `String` object, then the key becomes as tag and the value of object as value contained in the tag.
- If Object encountered in `genDetails` is a `Hashtable`, then it is sent into recursive mode, making the key as the parent tag for information contained in the `Hashtable`.
- If Object encountered in `genDetails` is a `Vector`, then the key becomes the parent tag and the singular form of the key becomes the children tag. If the `Vector` contains a `String`, then first rule is applied, and if it contains a `Hashtable`, second rule is applied.

The source code for method `populateContentHandler` can be checked at appendix B.1.3 under class `XMLTools`.

3.4.8 Eledge-ESUT

Eledge-ESUT is implementation of ESUT in Eledge Open Source Learning Management System developed by University of Utah. The `esutvariable` package is customized in this case also, to talk to the backend database. The backend database for Eledge is MySQL. The detailed source code for this implementation can be checked at appendix B.3 . Unlike CMS-ESUT, there are no restrictions on displaying the entire code. Like CMS-ESUT, we would show implementation of the method `processOutputXMLNewEvent(ByteArrayOutputStream, PrintWriter)` of class `EsutProcessOutputXML.java` showing how backend database is updated and show the implementation of method `getInputXML(HttpServletRequest)` from class `EsutXMLInputCreatorMain` to show how the information is queried from the database.

processOutputXMLNewEvent

The code can be checked at appendix B.3.2 under class `EsutProcessOutputXML`.

getInputXML

The code can be checked at appendix B.3.2 under class `EsutXMLInputCreatorMain`. The implementation of parse is same as that of its implementation in CMS.

It is evident from both CMS-ESUT and Eledge-ESUT that the code developed by implementing organization is nothing more than parsing an XML document and processing the database queries.

3.4.9 EduTools Evaluation

According to EduTools [26], all LMS applications can be classified into

- Learner Tools
- Support Tools

Learner Tools are further classified into three categories. They are

- Communication Tools
- Productivity Tools
- Student Involvement Tools

Support Tools are further classified into three categories. They are

- Administration
- Course Delivery
- Curriculum Design

Since all the tools mentioned above are based on request-response system, evaluating some extra tools would give us a better understanding of whether the ATO concept in general has an advantage over MTO/ETO approach of development. For the purpose of this thesis, we selected three tools, one each from Learner Tools category for evaluation. The tools thus selected are:

- File Exchange (Communication Tool)
- Bookmarks (Productivity Tool)
- Self Assessment Tool (Student Involvement Tool)

The detailed FP analysis of these tools is carried out in next chapter.

Chapter 4

DISCUSSION

This chapter discusses in details the results obtained from both Event Sign-Up Tool and EduTools applications. As mentioned earlier, there are two kind of metrics to measure code. They are Lines Of Code (LOC) and Function Point (FP) Analysis.

4.1 Code Sizing Metrics

- **Lines of Code (LOC):** As the name suggests, this method uses the length of the code as sizing metric. This is the most commonly used measure of source code program length [46]. However, this is not a very good measure of the complexity of the code. There are other issues such as handling “blank lines, comment lines, data declarations, and lines that contain several separate instructions” [46]. Care should be taken to ensure that counting is fair while comparing similar programs. For the purpose of this thesis, since all the code is developed by a single developer, this is a very good measure for measuring the efforts needed to write a particular program.
- **Function Points:** Function Point is broadly accepted as a measurement unit for the size of software. It is defined as “ a measurement of the functionality of the software product in standard units independent of the coding language. A Function Point is a measure of software functionality based on the counting or estimated number of “externals” (inputs, outputs, inquiries, and interfaces) of a system plus the estimated number of internal files of a program unit.” [27]. This metric is independent of the coding language and “since it measures the functional requirements, the measured size

stays constant despite the programming language, design technology, or development skills involved” [37]. For the purpose of this thesis, this method is used to calculate the complexity involved in writing the varying part vis-a-vis the whole program from the start. International Function Point Users Group (IFPUG) has developed a set of guidelines for assisting in calculating Function Points for an application. According to the guidelines [41], Function Points are calculated differently for

- Development Projects : These Function Points “measure the functionality provided to end users with the first installation of the application” [19]. These Function Points are used for projects which are yet to be developed.
- Enhancement Projects: These Function Points “measure modifications to existing applications and include the combined functionality provided to users by adding new functions, deleting old functions, and changing existing functions” [19].
- Application Projects: These Function Points “measure an installed application” [19]. These Function Points are used for the purpose of knowing the size of an already existing application. The Function Point calculations used in this thesis are of this type. Figure 4.1 describes the salient parameters needed to calculate Function Points.

The following steps are used to calculate Function Points.

- Application Boundary : This is a boundary defined for the application. “The application boundary is the border between the application being measured and either the external applications or the user domain” [19]. There are general guidelines about the boundary but there are no hard and fast rules for the definition of the boundary. It depends upon what user considers as his or her application boundary.
- External Inputs (EI) Identification: These are the inputs needed from an end-user to update or query the Internal Logical File. It consists of a typical input form fields, buttons, error messages. This is considered to be a transactional function. “An EI is an elementary process of the application that processes data or control information that enters from outside the boundary of the application” [19]. EI are mostly used to update the information in an Internal Logical File (ILF).

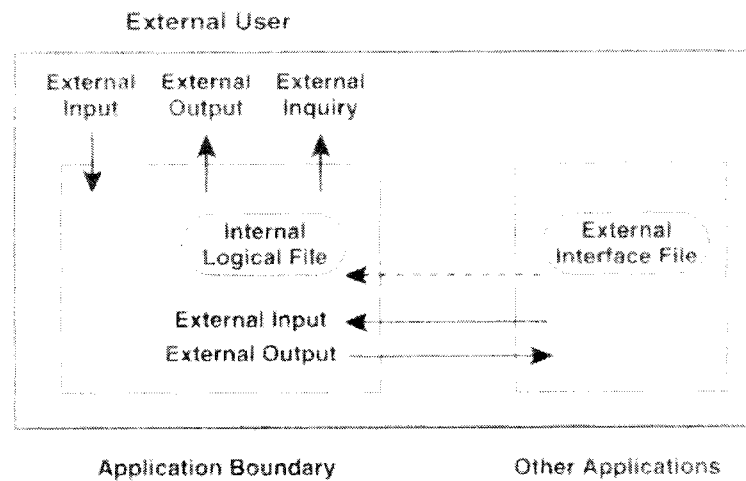


Figure 4.1: Functionality recognized in Function Point counting [19], by permission

- **External Outputs (EO) Identification:** This is a transaction function similar to EI but the data moves from inside the application boundary to outside. “An EO is an elementary process of the application that generates data or control information that exits the boundary of the application” [19]. The output generated must have at least one calculated value combining one or more fields to generate the resultant value.
- **External Inquiries (EQ) Identification:** This is a transaction function. “An EQ is an elementary process of the application that results in the retrieval of data or control information that is sent outside the application boundary” [19]. No ILF is maintained by this type of elements. They are used for fetching information from an ILF or an EIF.
- **Internal Logical Files (ILFs) Identification:** This is a data function. “An ILF is a user-identifiable group of logically related data or control information maintained within the boundary of the application” [19]. ILF are used for storing data related to the application. An ILF can be anything ranging from a database table to a simple text file.
- **External Interface Files (EIF) Identification:** This is also a data function similar to

ILF except that it is maintained outside the boundary of the application. “An EIF is a user-identifiable group of logically related data or control information referenced by the application but maintained within the boundary of a different application” [19].

- Unadjusted FP Calculation (UAF): There is a common table developed by IFPUG [41] for calculating the Unadjusted Function Point for development, enhancement and application projects. The table 4.1 shows the value associated with each component mentioned above. There are different rules for each component in establishing the components associated with it. The score calculation for EI data elements is shown in table 4.2. The File Type Referenced is either ILF or an EIF.

Once the UAF is calculated, it is multiplied with various factors to get the final FP calculations for each kind of project. However, for the purposes of this thesis, since all the projects being evaluated are application projects, the UAF has been used to demonstrate the effect of using ATO applications over traditional applications.

Table 4.1: UAF Calculation Table

Type of Component	Complexity of Components			Total
	Low	Average	High	
External Inputs	-- x 3 = --	-- x 4 = --	-- x 6 = --	
External Outputs	-- x 4 = --	-- x 5 = --	-- x 7 = --	
External Inquiries	-- x 3 = --	-- x 4 = --	-- x 6 = --	
Internal Logical Files	-- x 7 = --	-- x 10 = --	-- x 15 = --	
External Interface Files	-- x 5 = --	-- x 7 = --	-- x 10 = --	
Total Number of Unadjusted Function Points				

Table 4.2: EI Rating Table

Files Type Referenced (FTR)	Data Elements		
	1-4	5-15	>15
<2	Low(3)	Low(3)	Average(4)
2	Low(3)	Average(4)	High(6)
>2	Average(4)	High(6)	High(6)

4.2 LOC Vs FP

According to the IFPUG manual [41], FP is a better measure for software productivity because of the following reasons.

- The number of lines of code delivered is dependent upon the skill level of the programmer.
- Higher-level languages such as Forte, Dynasty, Visual Basic, JavaTM Script, or other visual languages require far fewer lines of code than Assembler, COBOL, or C to perform the same functionality. That is, there is an inverse relationship between the level of language and the work output (when work output is LOC).
- The actual number of LOC is not known until the project is almost completed. Therefore, LOC cannot be used to estimate the effort or schedule of a project. Function Points can be derived from requirements and analysis documents that are available early in a project life cycle.
- There is no agreed upon method to count lines of code. The statement and type of statements used in Visual C++, Assembler, COBOL, SQL are completely different from each other. Also, it is common for an application to have a combination of different languages as part of its code.

However, these contentions do not apply for the comparison of results from LOC and FP in our case. The reasons are explained below.

- Since the entire applications are developed by a single developer, the LOC comparison is quite acceptable ignoring the skill level of the programmer.

- The applications are developed in Java™ . Hence, there is no ambiguity about calculating the size of the code using LOC.
- The size of the application is calculated after the development of the applications and hence the values of LOC is precise.
- With respect to the programs in the application, the total number of lines in the code including comments, declarations, and so on are calculated, and hence is consistent with respect to the comparison.

4.3 Event Sign-Up Tool

This section evaluates the efforts needed to develop Event-Sign Up Tool with ATO approach as well as with ETO/MTO approach.

4.3.1 Database Structure Comparison

The database structure for both Eledge and CMS vary considerably. For example the tables are named differently and also some of the fields are named differently or do not even exist. For instance, the CourseID field does not exist in the Eledge database. Yet, using the ATO approach, the application is integrated into CMS, seamlessly, with only a small fraction of code being re-implemented.

Eledge Database Structure

ESUT in Eledge uses two tables. The screenshots for these tables are shown in figure 4.2 and figure 4.3.

CMS Database Structure

CMS also has two tables for ESUT. However, the tables are named differently and are called `esut_course_events` and `esut_course_sessions`. The screenshots for the tables are shown in figure 4.4 and figure 4.5.

Field	Type	Null	Key	Default	Extra
Posted	int(11)			0	
EventId	int(11)		PRI		auto_increment
StudentIDNumber	varchar(100)				
Title	varchar(100)				
Description	varchar(100)				
NoOfSessions	int(11)			0	
MaxChoices	int(11)			0	
EndDate	date			0000-00-00	

Figure 4.2: Eledge Events Table

Field	Type	Null	Key	Default	Extra
sessionno	int(11)			0	
eventid	int(11)			0	
title	varchar(100)				
noofseats	int(11)			0	
description	varchar(100)	YES			

Figure 4.3: Eledge Sessions Table

Column Name	Data Type	Length	Allow Nulls
Archived	int	4	✓
Archivaldatetime	datetime	8	✓
CourseEventID	int	4	
CourseID	int	4	✓
Name	varchar	400	✓
Description	varchar	8000	✓
NoOfSessions	int	4	✓
MaxChoices	int	4	✓
Enddate	datetime	8	✓
EventReleased	int	4	✓

Figure 4.4: CMS esut_course_events Table

Column Name	Data Type	Length	Allow Nulls
CourseSessionID	int	4	
CourseEventID	int	4	
Name	varchar	400	✓
Description	varchar	8000	✓
Ind	int	4	
Limit	int	4	✓

Figure 4.5: CMS esut_course_sessions Table

4.3.2 LOC Comparison

The LOC comparison is shown in table 4.3. It is evident from the table and the graph 4.6 that both CMS and Eledge needed less than 25% of the code developed in the basecode. It should also be noted that both CMS and Eledge contained framework implementation of esutvariable package, which should bring down the amount of lines used to even less than what is shown in the table and the graph. The basecode implemented only the framework for esutvariable package thereby showing lesser LOC as compared to the needs of a full fledged ESUT application.

4.3.3 Function Point Comparison

The Function Points needed to develop the whole application without using ATO is compared with the Function Points needed to develop only the esutvariable package. Since the Function Point metric is independent of the platforms, the Function Point calculations for Eledge and CMS will not vary much.

ESUT Function Point Calculation

The application has three forms. The detailed FP analysis for each form is shown below:

- ESUT Main Page: This form has recursive data information. This has
 - Recursive Fields: Date, Sessions, Choices, Title and Description (5).
 - Recursive Icons: Delete Event and Post Event (2).
 - URL: The URL for “New Event”(1).

Hence the total number of data elements for this form are 8 (5+2+1) and references 2 database tables. This would qualify for an average level form as per the IFPUG manual [41]. The score for an average level form is 4. Hence the total Unadjusted FP count for this form is $8*4 = 32$.

Table 4.3: LOC Comparison Table

Package esutfixed	Basecode LOC	Eledge LOC	CMS LOC*
EsutDeleteEvent.java	39		
EsutMain.java	49		
EsutNewEvent.java	83		
EsutNewSession.java	74		
EsutPostEvent.java	40		
EsutXMLInputCreatorNewSession.java	111		
EsutXMLOutputCreatorDeleteEvent.java	138		
EsutXMLOutputCreatorNewEvent.java	209		
EsutXMLOutputCreatorNewSession.java	200		
EsutXMLOutputCreatorPostEvent.java	137		
EsutUtilities.java	226		
Package esutvariable			
EsutProcessOutputXML.java	82	262	246
EsutXMLInputCreatorMain.java	48	151	133
EsutXMLInputCreatorNewEvent.java	37	102	99
Package javaxslt			
XMLProcessor.java	81		
XMLTools.java	86		
XSLT files			
createEvent.xslt	82		
createSession.xslt	69		
listEvents.xslt	90		
reEnterEventData.xslt	122		
reEnterSessionData.xslt	96		
Total LOC	2099	515	478

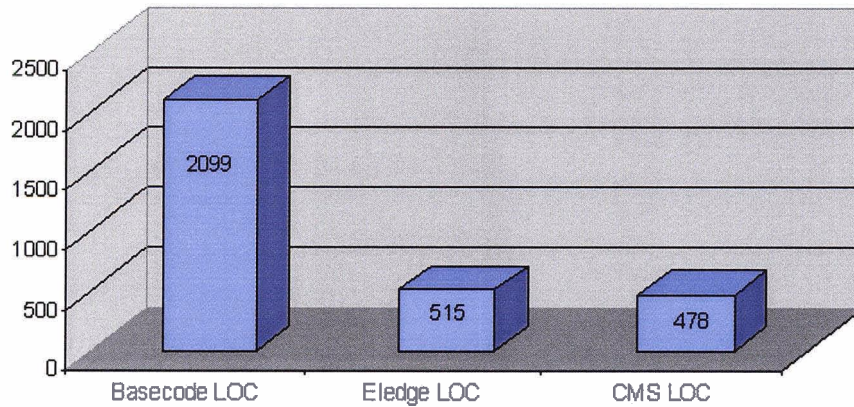


Figure 4.6: LOC Comparison Chart

- **Create New Event Page:** This page has 5 fields and 1 submit button and references only 1 table in the database. There are 12 error messages for this page which are counted as one data element. This would qualify as a low level form as per the IFPUG manual [41]. The score for a low level form is 3. Hence the total unadjusted FP count for this form is $7 \times 3 = 21$.
- **Create New Session(X of Y) Page:** This page has 3 fields and 1 submit button and this page references only 1 table in database. There are 4 error messages in this page. This page would also get 1 point for all the error messages. This would qualify as a low level form as per the IFPUG manual [41]. The score for a low level form is 3. Hence the total unadjusted FP count for this form is $5 \times 3 = 15$.

The application uses two database tables and hence qualifies for a low level score. The score of a low level ILF is 7. Hence the total unadjusted FP count for ILF is $7 \times 2 = 14$.

Based on these observations, the total score for the ESUT application is $32 + 21 + 15 + 14 = 82$.

esutvariable Package Function Point Calculation

The Function Point calculation for esutvariable can be divided based on the XML code that is required to generate the top pages. The calculation details are shown below.

- **ESUT Main Page:** This page has multiple fields but the processing for the fields is quite simple. Hence, this page, combining all the fields, is assigned 1 point. Another point is assigned for the confirmation/notification/error messages. This page references 2 database tables and hence would qualify as an average level form as per the IFPUG manual [41]. The score for a average level form is 4. Hence the total unadjusted FP count for this form is $2*4 = 8$.
- **Create New Event Page:** This page caters to 5 recursive fields. This form references only 1 table in the database. Similar to the ESUT Main Page, this page is also given 2 points, one for the fields and another for the notification/confirmation/error messages. This page also qualifies as a low level form as per the IFPUG manual [41]. The score for a low level form is 3. Hence the total unadjusted FP count for this form is $2*3 = 6$.
- **Create New Session (X of Y) page:** This page is very similar to the “Create New Event” page and hence has been assigned 2 points and would qualify as a low level form as per the IFPUG manual [41]. The total unadjusted FP count for this form is $2*3 = 6$.

This package also handles two tables and hence, similar to the ESUT application unadjusted FP count, gets 14 unadjusted FP counts.

The total unadjusted FP count for this package is $8+6+6+14 = 34$, which is much less than the FP count for the total application. It is clear from the 4.7 that the Function Point count for esutvariable package is about 60% less than that for the whole application.

4.4 EduTools

This section analyzes the tools selected from EduTools [26] website. The detailed code analysis of all the tools using both LOC and FP calculation is beyond the scope of this thesis, as the LOC comparison would need us to develop all the applications. Hence, we would analyze the above tools using Function Point analysis which does not need any development and is considered generally a better metric than LOC.

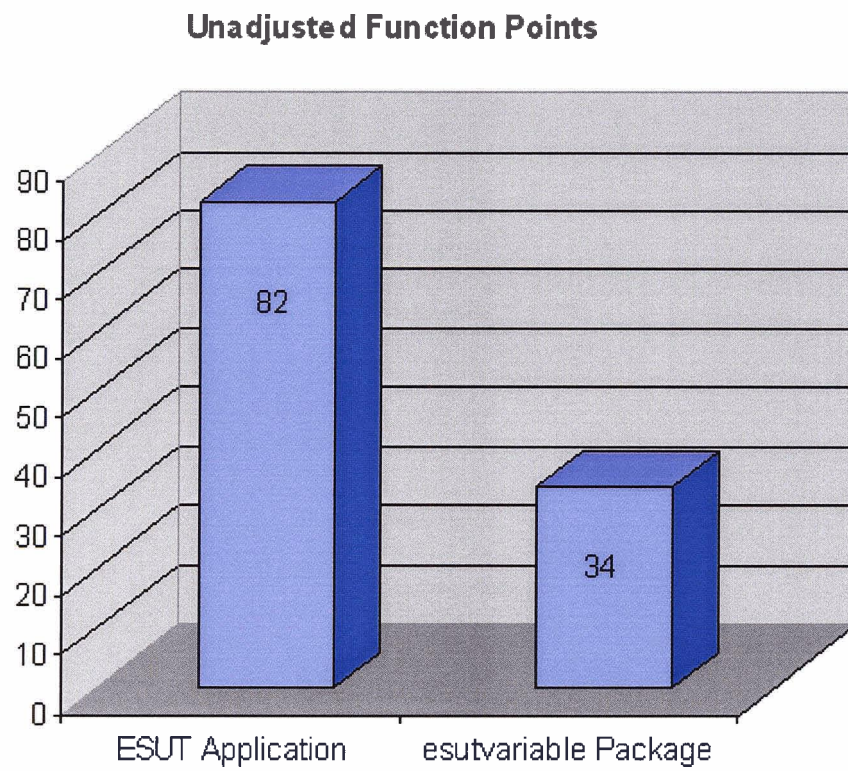


Figure 4.7: FP Comparison Chart

4.4.1 File Exchange Tool

According to EduTools site [26], “File exchange tools allow learners to upload files from their local computers and share these files with instructors or other students in an online course”.

The application consists of two forms

- Input Form
- Edit Form

Input Form

The input form will consist of following data fields

- File input field : For the file name that is to be uploaded.
- Shareable checkbox: A checkbox stating whether the file is shareable with other students or not.
- Browse button: For browsing the local directory.
- Add More Files link: This link will add another combination of File input field, Shareable checkbox and Browse button.
- Submit button: The button for submitting the file to the server.
- Error messages: Standard error messages.

The input form needs to communicate with 1 table for the information needed for this tool. The table will contains the following fields

- studentid: A text value that stores student id.
- courseid: A text value that stores course id.
- filename: A text value that stores filename.
- shareable: A boolean value for storing information about whether the file is shareable with other students or not.

As per IFPUG manual [41], the form has $6*3 = 18$ unadjusted FP and the table would correspond to $1*7=7$ unadjusted FP. Hence the total unadjusted FP for this form are 25.

Edit Form

The edit form contains following recursive fields

- Filename: Name of the file already uploaded to the server.
- Shareable checkboxes: The checkboxes that state whether the file is shareable with other students or not.
- Delete checkboxes: The checkboxes for deleting files.
- Submit button: The submit button.

This form also refers the same table as by Input Form. As per IFPUG manual [41] the form has $4*3 + 1*7 = 19$ unadjusted FP. Hence, the application has $25+19=44$ unadjusted FP.

Variable Function Point Calculations

The variable FP calculations would consist of following

- Input Form: Like mentioned earlier with esutvariable calculations, since the processing of the fields in this form becomes very simple, the fields are assigned 1 FP. Since the table accesses one database table, the number of unadjusted FP are $1*3 + 1*7 = 10$ FP
- Edit Form: This form would be very similar to Input form and would have $1*3 + 1*7 = 10$ unadjusted FP.

There are total $10+10=20$ adjusted variable FP. Hence we can conclude that the unadjusted FP for application (44) are much higher than the unadjusted FP for the application needed for the variable part (20).

4.4.2 Bookmarks

According to EduTools [26] site, “Bookmarks allow students to easily return to important pages within their course or outside their course on the web. In some cases bookmarks are for an individual students private use, and in others can be shared with an instructor or amongst an entire class”.

The application will consist of two forms

- Input Form
- Edit Form

Input Form

The input form consists of following data fields

- Title input field: Title of the bookmark.
- URL input field: URL of the bookmark.
- Share with faculty checkbox: A checkbox stating whether the bookmark is shareable with faculty.
- Share with students checkbox: A checkbox stating whether the bookmark is shareable with other students.
- Submit button: The button for submitting the input to the server.
- Error messages: Standard error messages.

The input form uses 1 table for the information needed for this tool. The table will contain the following fields

- studentid: As explained in File Exchange Tool above.
- courseid: As explained in File Exchange Tool above.
- bmtitle: Bookmark title.
- bmurl: Bookmark URL.

As per IFPUG manual [41], the form has $6*3 = 18$ unadjustable FP and the table would correspond to $1*7=7$ FP. Hence the total unadjusted FP for this form are 25.

Edit Form

The edit form contains following recursive fields

- Bookmark Title.
- Bookmark URL.
- Share with faculty checkbox.
- Share with students checkbox.
- Delete checkbox.
- Submit button (not recursive).

This form also would refer the same table as by Input Form. As per IFPUG manual [41], the form has $6*3 + 1*7 = 25$ unadjusted FP. Hence, the application has $25+25=50$ unadjusted FP.

Variable Function Point Calculations

- Input Form: With esutvariable calculations, the fields are assigned 1 FP. Since the table accesses one database table, the form has $1*3 + 1*7 = 10$ unadjusted FP.
- Edit Form: This form would be very similar to Input form and would have $1*3 + 1*7 = 10$ unadjusted FP.

Hence the variable part of the application has $10+10=20$ adjusted FP.

The unadjusted FP for application (50) are much higher than the unadjusted FP for the application needed for the variable part(20).

4.4.3 Self Assessment Tool

According to EduTools site, “Self-assessment tools allow students to take practice or review tests online. These assessments do not count toward a grade”.

The application consists of following forms

- Main Form.

- Add Question Form.
- Add Answer Form.
- Student Question Form.
- Student Answer Details Form.

Main Form

This form will consist of all the questions created by the instructors and should be able to edit or delete them. For the purpose of this thesis, we would not go into further details of the functionalities of this form.

The form will consist of following recursive fields and links

- Question input field: The input field for the question.
- Answer Type drop down: A drop down list with options like multiple answers, single answer etc.
- No of Answers input field: No of choices displayed to students.
- Delete Question link: Link to delete a question and all the answers.
- Edit Question link: To edit the question.
- Post Question link: To post the question to students. The question then goes to the question bank for the course.

In addition to above there will be error messages. The database will consist of 2 tables corresponding to the information needed for this tool.

Questions Table

This table will consist of following fields

- questionno: An integer value for the question number. Mostly auto-increment type.
- courseid: Same as explain above in File Exchange and Bookmarks tool.
- question: A text value that stores the question.

- noofanswers: An integer value containing information about number of choices given to students.
- multipleanswers: A boolean value containing information about whether the question has multiple answers or single answer.

Answers Table

This table will consist of following fields

- questionno: As explained above for Questions table.
- answerno: An integer value containing the serial number of the answer.
- rightanswer: A boolean value stating whether the answer is a right answer or not.
- answer: A text value containing answer.
- explanation: A text value explaining why an answer is right or wrong. It is an optional field.

As per IFPUG manual [41], since the form has 2 tables and 7 data elements, the form has $7*4 = 28$ unadjusted FP and the table would correspond to $2*7=14$ FP. Hence the total unadjusted FP for this form are $28+14=42$.

Add Question Form

The add question form contains following data elements

- Question input field: This field contains the question.
- Answer Type drop down field: A drop down containing options about whether the question has a single answer or multiple answers.
- No. Of Answers field: This field contains information about how many choices the students should get. This field would also decide about how many times the "Add Answer Form" explained below should be repeated.
- Submit button: A submit button.
- Error Messages: Standard error messages.

This form uses one table, Questions, which is mentioned above.

As per IFPUG manual [41], the form has $5*3 + 1*7 = 22$ unadjusted FP.

Add Answer Form

There will be multiple answer forms corresponding to every question. As explained above, the form gets repeated as many times as the input value in “No. Of Answers” input field above. This form has following data elements

- Answer Input field: This field contains the answer.
- Right Answer checkbox: This field contains the information whether it is a right answer or not. In case of single answer questions, once this checkbox is clicked, the checkbox will be disabled in subsequent forms.
- Detailed Reasoning textbox: This is a textbox containing information about why the answer is right or wrong.
- Submit button: A submit button.
- Error Messages: Standard error messages.

The form uses one database table, i.e; Answers. The details of this table are mentioned above.

As per IFPUG manual [41], the form has $5*3 + 1*7 = 22$ unadjusted FP.

Student Question Form

This form has following data elements

- Question: This field contains the question. This field can not be edited.
- Possible Answers: This is a recursive field containing all the answers. If the student has to choose one answer, this would contain a series of radio buttons, otherwise it would contain a series of checkboxes.
- Submit button: A submit button.
- Error Messages: Standard error messages.

This form refers to both Questions and Answers database.

As per IFPUG manual [41], the form has $4*4 + 2*7 = 30$ unadjusted FP

Student Answer Detail Form

This form has following data elements. None of the fields in this form can be edited.

- Question: This contains the question attempted by the student.
- Answer: This contains the student's answer.
- Answer Explanation: This explains why the answer was right or wrong.
- Next Question button: This buttons takes the student to next question.
- Quit button: This buttons quits the application.

This form also refers to both Questions and Answers table. As per IFPUG manual [41], the form has $5*4 + 2*7 = 34$ unadjusted FP. Hence the application has $42+22+22+30+34 = 150$ unadjusted FP.

Variable Function Point Calculations

- Main Form: With esutvariable calculations, the fields are assigned 1 FP. Since the table accesses two database table, the number of unadjusted FP are $1*4 + 2*7 = 18$ FP.
- Add Question Form: This form has 5 fields and accesses 1 database table. Therefore, the form has $1*3 + 1*7 = 10$ unadjusted FP.
- Add Answer Form: This is very similar to Add Question Form and thus has 10 unadjusted FP.
- Student Question Form: This is similar to Main Form and thus has 18 unadjusted FP.
- Student Answer Form: This is also similar to Main Form and thus has 18 unadjusted FP.

Hence there are total $18+10+10+18+18=74$ adjusted variable FP. Thus we can conclude that the unadjusted FP for application (150) are much higher than the unadjusted FP for the needed for the variable part(74) of the application.

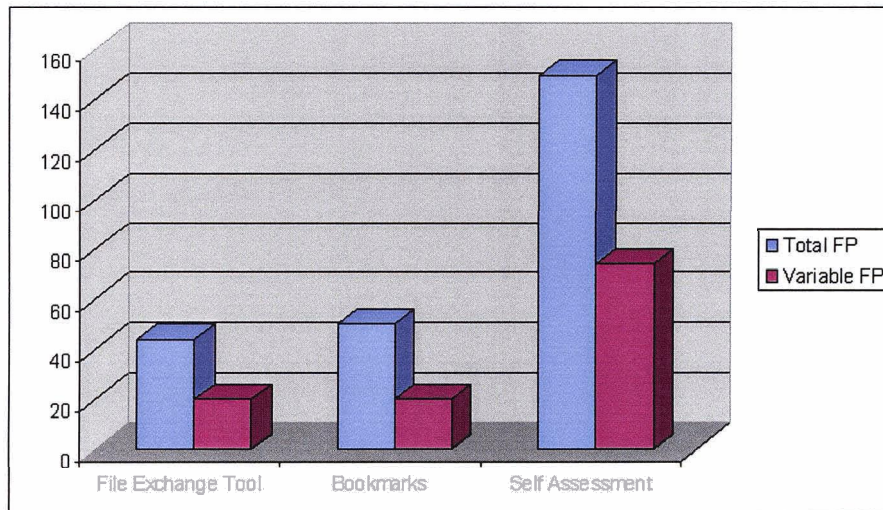


Figure 4.8: Function Point Comparison for EduTools Applications

4.4.4 EduTools FP Analysis

We observed that the efforts needed to develop the variable part is much lower than the efforts needed to develop the full application from start. The table 4.4 shows the summary of FP needed to develop the application and the variable part. The figure 4.8 also shows the results graphically.

Table 4.4: EduTools FP Comparison Table

Application	Total FP	Variable FP	Variabe/Total %
File Exchange Tool	44	20	45
Bookmarks	50	20	40
Self Assessment	150	74	49
Total	244	114	47

Chapter 5

CONCLUSION, LIMITATIONS AND FURTHER RESEARCH

The thesis explored the possibility of extending the ATO concept used in industrial production to software development in educational environments. The thesis also emphasized that the software development process is not very different from industrial production and thus we can learn to develop better applications using our manufacturing expertise. Through this thesis, we demonstrated that it is possible to extend our experience in industrial production to develop re-usable LMS applications thereby sharing educational experience of an institution with other institutions.

We observed that the ATO approach for developing LMS applications is better than the traditional ETO/MTO approach for following reasons:

- The applications developed have already been tested and in most cases even implemented by one or more institutions. The applications thus are more robust and scalable than a traditional application developed using MTO or ETO approach.
- Since the applications are developed using open source technologies based on XML and XSLT, the presentation layer is separated from the business logic. This gives the flexibility to the implementing institution to even change the user interface (look and feel) of the applications to match their standard web templates or user interfaces.
- The learning experience gained by an institution implementing an application can be shared with other institutions at much lesser cost, thereby enabling the easy flow of

information from one institution to another.

5.1 Limitations

The concept has following limitations.

- The ATO concept is useful only if the application is going to be implemented at more than one LMS.
- Like every other new concept, there is time associated with learning curve.
- The technologies used are complicated when compared with standard servlet and database communication.
- The concept is tested only with J2EE based technologies. The success of the concept in another technologies is not tested. However, since FP analysis is used for comparison, the result is expected to be similar to one observed above.

5.2 Further Research

Further research can be carried out in the following areas to extend the research carried out in this thesis.

- Developing specifications or standards for developing applications using the ATO approach.
- A set of API can be developed to assist developers in handling XML inputs and outputs. Though there are already API available for the purpose, special API can cater specifically to the concept of ATO development approach.
- A repository like repositories used for sharing learning objects or a software repository like SOURCEFORGE [53] can be developed to share the applications developed using the ATO approach. The institutions can check in those repositories, if these repositories already have an ATO application available they need or can make use of, rather than developing it from scratch.
- Research needs to be carried out in extending the ATO concept between various technology platforms like .NET and J2EE.

Appendix A

XML Schemas and XSLT Files

A.1 XML Schemas

The following xml schemas are used in the thesis. They all have .xsd extension.

A.1.1 DeleteEvent

```
<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!--DeleteEvent.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ESUT">
    <xs:complexType>
      <xs:all>
        <xs:element ref="eventId"/>
        <xs:element ref="role"/>
        <xs:element ref="username"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="eventId" type="xs:byte"/>
```



```

<xs:element name="role">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="instructor"/>
      <xs:enumeration value="student"/>
      <xs:enumeration value="TA"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.2 ListEvents

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!-- ListEvents.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ESUT">
    <xs:complexType>
      <xs:all>
        <xs:element ref="events"/>
        <xs:element ref="courseId"/>
        <xs:element ref="role"/>
        <xs:element ref="username"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="choices" type="xs:int"/>
  <xs:element name="courseId" type="xs:string"/>
  <xs:element name="description" type="xs:string"/>
  <xs:element name="endDate">

```

```
<xs:simpleType>
  <xs:restriction base="xs:date"/>
</xs:simpleType>
</xs:element>
<xs:element name="event">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="status"/>
      <xs:element ref="eventId"/>
      <xs:element ref="description"/>
      <xs:element ref="endDate"/>
      <xs:element ref="title"/>
      <xs:element ref="choices"/>
      <xs:element ref="sessions"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="eventId" type="xs:int"/>

<xs:element name="events">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="event" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
<xs:element name="role">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="teacher"/>
      <xs:enumeration value="student"/>
      <xs:enumeration value="TA"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
```

```

    </xs:simpleType>
</xs:element>
<xs:element name="sessions" type="xs:int"/>
<xs:element name="status">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="complete"/>
      <xs:enumeration value="incomplete"/>
      <xs:enumeration value="posted"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="title" type="xs:string"/>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.3 NewEvent

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!-- NewEvent.xsd --> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ESUT">
    <xs:complexType>
      <xs:all>
        <xs:element ref="username"/>
        <xs:element ref="role"/>
        <xs:element name="courseId" type="xs:string"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="description" type="xs:string"/>

```

```

<xs:element name="endDate" type="xs:date"/>
<xs:element name="maxChoices" type="xs:byte"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="noOfSessions" type="xs:byte"/>
<xs:element name="role">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="instructor"/>
      <xs:enumeration value="student"/>
      <xs:enumeration value="TA"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.4 NewSession

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!--NewSession.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ESUT">
    <xs:complexType>
      <xs:all>
        <xs:element ref="eventId"/>
        <xs:element ref="noOfSeats"/>
        <xs:element ref="sessionNo"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="description" type="xs:string"/>

```

```

<xs:element name="eventId" type="xs:byte"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="noOfSeats" type="xs:byte"/>
<xs:element name="role">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="instructor"/>
      <xs:enumeration value="TA"/>
      <xs:enumeration value="student"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="sessionNo" type="xs:byte"/>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.5 PostEvent

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!--PostEvent.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ESUT">
    <xs:complexType>
      <xs:all>
        <xs:element ref="eventId"/>
        <xs:element ref="role"/>
        <xs:element ref="username"/>
      </xs:all>
    </xs:complexType>
  </xs:element>
  <xs:element name="eventId" type="xs:byte"/>

```

```

<xs:element name="role">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="instructor"/>
      <xs:enumeration value="student"/>
      <xs:enumeration value="TA"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.6 ReEnterEventData

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!--ReEnterEventData.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ESUT">
    <xs:complexType>
      <xs:all>
        <xs:element name="errors">
          <xs:complexType>
            <xs:sequence>
<xs:element name="error" type="xs:string" maxOccurs="unbounded"/>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
        <xs:element ref="sessions"/>
        <xs:element ref="choices"/>
        <xs:element ref="endDate"/>
        <xs:element ref="title"/>

```

```
        <xs:element ref="description"/>
        <xs:element ref="courseId"/>
        <xs:element ref="role"/>
        <xs:element ref="username"/>
    </xs:all>
</xs:complexType>
</xs:element>
<xs:element name="choices" type="xs:int"/>
<xs:element name="courseId" type="xs:string"/>
<xs:element name="description" type="xs:string"/>
<xs:element name="endDate">
    <xs:simpleType>
        <xs:restriction base="xs:date"/>
    </xs:simpleType>
</xs:element>
<xs:element name="event">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="eventId" type="xs:int"/>
<xs:element name="events">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="event" maxOccurs="unbounded"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="role">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="teacher"/>
            <xs:enumeration value="student"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
```

```

        <xs:enumeration value="TA"/>
    </xs:restriction>
</xs:simpleType>
</xs:element>
<xs:element name="sessions" type="xs:int" />

<xs:element name="status">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="complete"/>
            <xs:enumeration value="incomplete"/>
            <xs:enumeration value="posted"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="title" type="xs:string"/>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.7 SaveEvent

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!--SaveEvent.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <xs:element name="ESUT">
        <xs:complexType>
            <xs:all>
                <xs:element ref="name"/>
                <xs:element ref="description"/>
                <xs:element ref="maxChoices"/>
                <xs:element ref="noOfSessions"/>
            </xs:all>
        </xs:complexType>
    </xs:element>
</xs:schema>

```



```

        <xs:element ref="endDate"/>
        <xs:element ref="username"/>
        <xs:element ref="role"/>
        <xs:element name="courseId" type="xs:string"/>
    </xs:all>
</xs:complexType>
</xs:element>
<xs:element name="description" type="xs:string"/>
<xs:element name="endDate" type="xs:date"/>
<xs:element name="maxChoices" type="xs:byte"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="noOfSessions" type="xs:byte"/>
<xs:element name="role">
    <xs:simpleType>
        <xs:restriction base="xs:string">
            <xs:enumeration value="instructor"/>
            <xs:enumeration value="student"/>
            <xs:enumeration value="TA"/>
        </xs:restriction>
    </xs:simpleType>
</xs:element>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.1.8 SaveSession

```

<?xml version="1.0" encoding="UTF-8"?> <!-- edited with XMLSPY v5
rel. 4 U (http://www.xmlspy.com) by Baljeet Dhaliwal (Simon Fraser
University) --> <!--W3C Schema generated by XMLSPY v5 rel. 4 U
(http://www.xmlspy.com)--> <!--SaveSession.xsd--> <xs:schema
xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
    <xs:element name="ESUT">
        <xs:complexType>

```

```

    <xs:all>
      <xs:element ref="eventId"/>
      <xs:element ref="name"/>
      <xs:element ref="description"/>
      <xs:element ref="noOfSeats"/>
      <xs:element ref="sessionNo"/>
      <xs:element ref="username"/>
      <xs:element ref="role"/>
    </xs:all>
  </xs:complexType>
</xs:element>
<xs:element name="description" type="xs:string"/>
<xs:element name="eventId" type="xs:byte"/>
<xs:element name="name" type="xs:string"/>
<xs:element name="noOfSeats" type="xs:byte"/>
<xs:element name="role">
  <xs:simpleType>
    <xs:restriction base="xs:string">
      <xs:enumeration value="instructor"/>
      <xs:enumeration value="TA"/>
      <xs:enumeration value="student"/>
    </xs:restriction>
  </xs:simpleType>
</xs:element>
<xs:element name="sessionNo" type="xs:byte"/>
<xs:element name="username" type="xs:string"/>
</xs:schema>

```

A.2 XSLT Files

The following xslt files are used in the thesis. They all have .xslt extension.

A.2.1 createEvent

```

<?xml version="1.0" encoding="UTF-8"?> <!--createEvent.xslt-->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:output method="html" version="1.0" encoding="UTF-8"
indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Event Signup Tool</title>
        <link rel="stylesheet" type="text/css"
href="../xslt/stylesheet.css"/>
      </head>
      <body>
        <div id="content">
          <h3> Create New Event </h3>
          <form id="createSessionForm" method="post"
action="EsutNewEvent" >
            <table align="center" width="500">
              <tr>
                <td align="left" colspan="2"></td>
              </tr>
              <tr>
                <td align="left" class="noDeco"></td>
                <td/>
              </tr>
              <tr>
                <td class="createEventLeftCol">
<input type="text" name=
"name" size="30"/>
                </td>
                <td class="createEventRightCol">

```

```

        Event Name
    </td>
</tr>
<tr>
    <td class="createEventLeftCol">
<textarea name="description"
cols="23" rows="10"/>
    </td>
    <td class="createEventRightCol">
        Event Description
    </td>
</tr>
<tr>
    <td class="createEventLeftCol">
<input type="text" name=
"noOfSessions" size="30"/>
<br/>
    </td>
    <td class="createEventRightCol">
        No. of Sessions
    </td>
</tr>
<tr>
    <td class="createEventLeftCol">
<input type="text" name=
"maxChoices" size="30"/>
    </td>
    <td class="createEventRightCol">
        No. of Choices
    </td>
</tr>
<tr>
    <td class="createEventLeftCol">

```

```

        <input type="text" name=
        "endDate" size="30"/><br/>
            </td>
            <td class="createEventRightCol">
                Signup End Date (yyyy-mm-dd)
            </td>
        </tr>
        <tr>
            <td class="noDeco" align="center"
            colspan="2">
                <input type="submit" value="Create"/>
            </td>
        </tr>
    </table>
    <input type="hidden" name="createEventForm"
    value="createEventForm"/>
    </form>
    </div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

A.2.2 createSession

```

<?xml version="1.0" encoding="UTF-8"?> <!--createSession.xslt-->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html" version="1.0" encoding="UTF-8"
    indent="yes"/>
    <xsl:template match="/">
        <html>
            <head>
                <title>Session Signup Tool</title>
            </head>
        </html>
    </template>
</xsl:stylesheet>

```

```

<link rel="stylesheet" type="text/css" href=
"../xslt/stylesheet.css"/>
</head>
<body>
  <div id="content">
    <h3> Create New Session (
      <xsl:value-of select="ESUT/sessionNo" /> of
      <xsl:value-of select="ESUT/noOfSessions" /> )

    </h3>
    <form method="post" action="EsutNewSession" >
      <table align="center" width="500">
        <tr>
          <td align="left" colspan="2"></td>
        </tr>
        <tr>
          <td align="left" class="noDeco"></td>
          <td/>
        </tr>
        <tr>
          <td class="createEventLeftCol">
<input type="text" name=
"name" size="30"/>
          </td>
          <td class="createEventRightCol">
            Session Name
          </td>
        </tr>
        <tr>
          <td class="createEventLeftCol">
<textarea name="description"
cols="23" rows="10"/>
          </td>

```

```

                <td class="createEventRightCol">
                    Session Description
                </td>
            </tr>
            <tr>
                <td class="createEventLeftCol">
                    <input type="text" name=
                    "noOfSeats" size="30"/><br/>
                </td>
                <td class="createEventRightCol">
                    No. of Seats
                </td>
            </tr>
            <tr>
                <td class="noDeco" align="center"
                colspan="2">
                    <input type="submit"
                    value="Create"/>
                </td>
            </tr>
        </table>
        <input type="hidden" name="createSessionForm"
        value="createSessionForm"/>
    </form>
</div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

A.2.3 listEvents

```

<?xml version="1.0" encoding="UTF-8"?> <!--listEvents.xslt-->
<xsl:stylesheet version="1.0"

```

```

xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
  <xsl:output method="html" version="1.0" encoding="UTF-8"
  indent="yes"/>
  <xsl:template match="/">
    <html>
      <head>
        <title>Event Signup Tool</title>
<link rel="stylesheet" type="text/css" href=
"../xslt/stylesheet.css"/>
      </head>
      <body>
        <div id="container">
          <div class="roundcont">
            <div class="roundtop">

            </div>
            <!-- end rounded corner code-->
            <div id="banner">
<span class="bannertext">Event Signup
</span> | Home |
<a href="EsutNewEvent">NEW EVENT</a> |
<a href="CourseEventHelp.htm">Help</a>
              </div>
            <div id="content">
              <xsl:apply-templates select="ESUT"/>
            </div>
          </div>

          <div class="roundbottom">
<img src="../images/bl.gif" alt="" width="15"
height="15" class="corner"

```



```

        style="display: none"/>
      </div>
    </body>
  </html>
</xsl:template>
  <xsl:template match="ESUT">
<table width="100%" border="0" align="center"
cellpadding="10" cellspacing="1">
  <tr>
    <td width="75" class="eventtitle">Date</td>
    <td width="75" class="eventtitle">Sessions</td>
    <td width="75" class="eventtitle">Choices</td>
    <td class="eventtitle">Title</td>
    <td class="eventtitle" width="250">Description</td>
    <td width="139" colspan="3" class="eventtitle">
      Actions</td>
  </tr>
  <xsl:for-each select="events/event">
    <tr class="live">
      <td width="75" class="eventlive">
        <xsl:value-of select="endDate"/>
      </td>
      <td width="75" class="eventlive">
        <xsl:value-of select="sessions"/>
      </td>
      <td width="75" class="eventlive">
        <xsl:value-of select="choices"/>
      </td>
      <td width="250" class="eventlive">
        <a href="link">
          <xsl:value-of select="title"/>
        </a>
      </td>
    </tr>
  </xsl:for-each>
</table>

```

```

        <td class="eventlive">
            <xsl:value-of select="description"/>
        </td>
        <xsl:choose>
            <xsl:when test="status='posted'
            or status='incomplete'">
                <td width="33" class="eventtext">

                </td>
            </xsl:when>
            <xsl:when test="status='complete'">
                <td width="33" class="eventtext">
                    <a><xsl:attribute name="href">
EsutPostEvent?eventId=
<xsl:value-of select=
"eventId"/>
                    </xsl:attribute>
                        
                    </a>
                </td>
            </xsl:when>
        </xsl:choose>
        <td width="33" class="eventtext">
            <a>
                <xsl:attribute name="href">
EsutDeleteEvent?eventId=<xsl:value-of
select="eventId"/>
                </xsl:attribute>
                
            </a>

```

```

                </td>
            </tr>
        </xsl:for-each>
    </table>
</xsl:template>
</xsl:stylesheet>

```

A.2.4 reEnterEventData

```

<?xml version="1.0" encoding="UTF-8"?> <!--reEnterEventData.xslt-->
<xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html" version="1.0" encoding="UTF-8"
indent="yes"/>
    <xsl:template match="/">
        <html>
            <head>
                <title>Event Signup Tool</title>
<link rel="stylesheet" type="text/css" href=
"../xslt/stylessheet.css"/>
            </head>
            <body>
                <div id="content">
                    <h3> Create New Event </h3>
                    <ul>
                        <xsl:for-each select="ESUT/errors/error">
                            <li><xsl:value-of select="." /></li>
                        </xsl:for-each>
                    </ul>

                    <form method="post" action="EsutNewEvent" >
                        <table align="center" width="500">
                            <tr>
                                <td align="left" colspan="2"></td>
                            </tr>

```

```

        <tr>
            <td align="left" class="noDeco"></td>
        </tr>
        <tr>
            <td class="createEventLeftCol">
                <input type="text"
                name="name" size="30">
                    <xsl:attribute name="value">
                        <xsl:value-of
                        select="ESUT/name" />
                    </xsl:attribute>
                </input>
            </td>
            <td class="createEventRightCol">
                Event Name
            </td>
        </tr>
        <tr>
            <td class="createEventLeftCol">
                <textarea name="description"
                cols="23" rows="10">
                    <xsl:value-of
                    select=
                    "ESUT/description" />
                </textarea>
            </td>
            <td class="createEventRightCol">
                Event Description
            </td>
        </tr>
        <tr>
            <td class="createEventLeftCol">

```

```

        <input type="text"
            name="noOfSessions" size="30">
            <xsl:attribute
                name="value">
            <xsl:value-of
                select=
                "ESUT/noOfSessions"/>
        </input><br/>
    </td>
    <td class="createEventRightCol">
        No. of Sessions
    </td>
</tr>
<tr>
    <td class="createEventLeftCol">
        <input type="text" name=
            "maxChoices" size="30">
        <xsl:attribute
            name="value">
        <xsl:value-of
            select=
            "ESUT/maxChoices" />
        </xsl:attribute>
        </input>
    </td>
    <td class="createEventRightCol">
        No. of Choices
    </td>
</tr>
<tr>
    <td class="createEventLeftCol">
        <input type="text" name=
            "endDate" size="30">

```

```

        <xsl:attribute name="value">
        <xsl:value-of
            select="ESUT/endDate" />
        </xsl:attribute>
    </input><br/>
</td>

        <td class="createEventRightCol">
            Signup End Date (yyyy-mm-dd)
        </td>
    </tr>
    <xsl:apply-templates select="ESUT"/>
    <tr>
<td class="noDeco" align="center"
colspan="2">
    <input type="submit"
        value="Create"/>
</td>
</tr>
</table>
<input type="hidden" name="createEventForm"
value="createEventForm"/>
    </form>
    </div>
    </body>
</html>
</xsl:template>
</xsl:stylesheet>

```

A.2.5 reEnterSessionData

```

<?xml version="1.0" encoding="UTF-8"?>
<!--reEnterSessionData.xslt--> <xsl:stylesheet version="1.0"
xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
    <xsl:output method="html" version="1.0" encoding="UTF-8"

```



```

        <xsl:value-of
        select="ESUT/name"/>
    </xsl:attribute>
</input>
</td>
<td class="createEventRightCol">
    Event Name
</td>
</tr>
<tr>
<td class="createEventLeftCol">
    <textarea name="description"
    cols="23" rows="10">
<xsl:value-of
select="ESUT/description" />
    </textarea>
</td>
<td class="createEventRightCol">
    Event Description
</td>
</tr>
<tr>
<td class="createEventLeftCol">
    <input type="text" name=
    "noOfSessions" size="30">
    <xsl:attribute name="value">
    <xsl:value-of select=
    "ESUT/noOfSeats" />
    </xsl:attribute>
    </input> <br/>
</td>
<td class="createEventRightCol">
    No. of Sessions

```



```
</td>
      </tr>
      <tr>
        <td class="noDeco" align="center"
            colspan="2">
          <input type="submit"
                value="Create"/>
        </td>
      </tr>
    </table>
    <input type="hidden" name="createEventForm"
          value="createEventForm"/>
      </form>
    </div>
  </body>
</html>
</xsl:template>
</xsl:stylesheet>
```

Appendix B

Source Code

B.1 ATO-ESUT-basecode

The following packages are used in ATO-ESUT-basecode.

B.1.1 Package ca.sfu.elinc.esutfixed

`EsutDeleteEvent`

```
package ca.sfu.elinc.esutfixed;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import ca.sfu.elinc.esutvariable.*;
import ca.sfu.elinc.esutfixed.util.EsutUtilities;
import ca.sfu.elinc.esutfixed.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
```

```
/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */

public class EsutDeleteEvent extends HttpServlet{

    public void doGet(HttpServletRequest req, HttpServletResponse
    res) throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        ByteArrayOutputStream outputXML =
        EsutXMLOutputCreatorDeleteEvent.getOutputXML(req);
        if (EsutProcessOutputXML.
        processOutputXMLDeleteEvent(outputXML, out)){
            req.getRequestDispatcher("EsutMain").forward(req, res);
        }
    }
}
```

EsutMainEvent

```
package ca.sfu.elinc.esutfixed;

import java.io.*;
import javax.servlet.http.*;
import javax.servlet.*;
import java.util.*;
import ca.sfu.elinc.esutfixed.util.EsutUtilities;
import ca.sfu.elinc.esutvariable.*;
import ca.sfu.elinc.javaxslt.*;
/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
```

```
*
*/
public class EsutMain extends HttpServlet {

    public void doGet(HttpServletRequest req, HttpServletResponse
res)throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        ByteArrayOutputStream inputXML =
EsutXMLInputCreatorMain.getInputXML(req);
        String inX = inputXML.toString();
        String username = inX.substring
(inX.indexOf("<username>")+10, inX.indexOf("</username>"));
        String role = inX.substring(inX.indexOf("<role>")+6,
inX.indexOf("</role>"));
        String courseId = inX.substring(inX.indexOf("<courseId>")+10,
inX.indexOf("</courseId>"));
        HttpSession session = req.getSession(true);
        session.setAttribute("esut.username", username);
        session.setAttribute("esut.role", role);
        session.setAttribute("esut.courseId", courseId);
XMLProcessor sim = new XMLProcessor();
String uri = req.getRequestURI() ;
String path = getServletConfig().
getServletContext().getRealPath("xslt");
String xsltFileName = path+"/listEvents.xslt";
File f = new File (xsltFileName);
ByteArrayOutputStream baos2 = sim.
changeFormat(out,inputXML,new File(xsltFileName));
out.println(baos2);
    }

    public void doPost(HttpServletRequest req, HttpServletResponse
```

```
        res)throws ServletException, IOException {
            doGet(req, res);
        }
    }
}
```

EsutNewEvent

```
package ca.sfu.elinc.esutfixed;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import ca.sfu.elinc.esutvariable.*;
import ca.sfu.elinc.esutfixed.util.EsutUtilities;
import ca.sfu.elinc.javaxslt.*;
import ca.sfu.elinc.esutfixed.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutNewEvent extends HttpServlet{

    public void doGet(HttpServletRequest req, HttpServletResponse
res)throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        ByteArrayOutputStream inputXML =
```

```
        EsutXMLInputCreatorNewEvent.getInputXML(req);
        out.println(inputXML);
XMLProcessor sim = new XMLProcessor();
String uri = req.getRequestURI() ;
String path = getServletConfig().
getServletContext().getRealPath("xslt");
String xsltFileName = path+"/createEvent.xslt";
File f = new File (xsltFileName);
ByteArrayOutputStream baos2 = sim.
changeFormat(out,inputXML,new File(xsltFileName));
out.println(baos2);
    }

    public void doPost(HttpServletRequest req, HttpServletResponse
res)throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        Vector errors = EsutUtilities.validateNewEventParameters(req);
        if (errors.size(>0){//there are errors
            ByteArrayOutputStream errorXML =
            EsutXMLOutputCreatorNewEvent.getErrorXML(req, errors);
            XMLProcessor sim = new XMLProcessor();
            String uri = req.getRequestURI() ;
            String path = getServletConfig().
            getServletContext().getRealPath("xslt");
            String xsltFileName = path+"/reEnterEventData.xslt";
            File f = new File (xsltFileName);
            ByteArrayOutputStream baos2 = sim.
            changeFormat(out,errorXML,new File(xsltFileName));
            out.println(baos2);
        }
        else {
            ByteArrayOutputStream outputXML =
```

```

    EsutXMLOutputCreatorNewEvent.getSuccessXML(req);
    int eventId = EsutProcessOutputXML.
processOutputXMLNewEvent(outputXML, out);
    if (eventId>0){
        HttpSession session = req.getSession(true);
        String noOfSessions =
        (String)req.getParameter("noOfSessions");
        session.setAttribute
        ("esut.noOfSessions", noOfSessions);
        session.setAttribute("esut.sessionNo", "1");
        session.setAttribute("esut.eventId", ""+eventId);
        ByteArrayOutputStream inputXML =
        EsutXMLInputCreatorNewSession.getInputXML(req);

        XMLProcessor sim = new XMLProcessor();
        String uri = req.getRequestURI() ;
        String path = getServletConfig().getServletContext().
        getRealPath("xslt");
        String xsltFileName = path+"/createSession.xslt";
        File f = new File (xsltFileName);
        ByteArrayOutputStream baos2 = sim.
        changeFormat(out,inputXML,new File(xsltFileName));
        out.println(baos2);
    }
}
}
}
}

```

EsutNewSession

```

package ca.sfu.elinc.esutfixed;

import java.io.*;
import javax.servlet.*;

```

```
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import ca.sfu.elinc.esutvariable.*;
import ca.sfu.elinc.esutfixed.util.EsutUtilities;
import ca.sfu.elinc.javaxslt.*;
import ca.sfu.elinc.esutfixed.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutNewSession extends HttpServlet{

    public void doPost(HttpServletRequest req, HttpServletResponse
    res)throws ServletException, IOException {
        res.setContentType("text/html");
        PrintWriter out = res.getWriter();
        Vector errors = EsutUtilities.
        validateNewSessionParameters(req);
        if (errors.size()!=0){
            ByteArrayOutputStream errorXML =
            EsutXMLOutputCreatorNewSession.getErrorXML(req, errors);
            XMLProcessor sim = new XMLProcessor();
            String uri = req.getRequestURI() ;
            String path = getServletConfig().getServletContext().
            getRealPath("xslt");
```



```
String xsltFileName = path+"/reEnterSessionData.xslt";
File f = new File (xsltFileName);
ByteArrayOutputStream baos2 = sim.
changeFormat(out,errorXML,new File(xsltFileName));
out.println(baos2);
}
else {
    ByteArrayOutputStream outputXML =
    EsutXMLOutputCreatorNewSession.getSuccessXML(req);
    if (EsutProcessOutputXML.
processOutputXMLNewSession(outputXML, out)){
        HttpSession session = req.getSession(true);
        String noOfSess = (String)session.
getAttribute("esut.noOfSessions");
        int noOfSessions = (new Integer(noOfSess)).intValue();
        String sessionNo = (String)session.
getAttribute("esut.sessionNo");
        int newSessionNo = new Integer(sessionNo).
intValue() +1 ;
        if (newSessionNo<=noOfSessions){
            session.setAttribute
            ("esut.sessionNo", ""+newSessionNo);
            ByteArrayOutputStream inputXML =
            EsutXMLInputCreatorNewSession.getInputXML(req);
            XMLProcessor sim = new XMLProcessor();
            String uri = req.getRequestURI() ;
            String path = getServletConfig().
getServletContext().getRealPath("xslt");
            String xsltFileName = path+"/createSession.xslt";
            File f = new File (xsltFileName);
            ByteArrayOutputStream baos2 =
            sim.changeFormat(out,inputXML,
            new File(xsltFileName));
```

```
        out.println(baos2);
    }
    else {
        req.getRequestDispatcher("EsutMain").
        forward(req, res);
    }
}
}
}
```

EsutPostEvent

```
package ca.sfu.elinc.esutfixed;

import java.io.*;
import javax.servlet.*;
import javax.servlet.http.*;
import java.sql.*;
import java.util.*;
import ca.sfu.elinc.javaxslt.XMLProcessor;
import ca.sfu.elinc.esutvariable.*;
import ca.sfu.elinc.esutfixed.util.EsutUtilities;
import ca.sfu.elinc.esutfixed.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutPostEvent extends HttpServlet{
```

```
public void doGet(HttpServletRequest req, HttpServletResponse
res)throws ServletException, IOException {
    res.setContentType("text/html");
    PrintWriter out = res.getWriter();
    ByteArrayOutputStream outputXML =
    EsutXMLOutputCreatorPostEvent.getOutputXML(req);
    if (EsutProcessOutputXML.processOutputXMLPostEvent
(outputXML, out)){
        req.getRequestDispatcher("EsutMain").forward(req, res);
    }
}
}
```

EsutXMLInputCreatorNewSession

```
package ca.sfu.elinc.esutfixed;

import java.io.*;
import java.sql.*;
import javax.servlet.http.*;
import java.util.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;

/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
```

```
*
*/
public class EsutXMLInputCreatorNewSession extends AbstractXMLReader {

    private static final Attributes EMPTY_ATTR = new AttributesImpl();
    public void parse(InputSource input){
        try{
            ContentHandler ch = getContentHandler();
            if (ch == null) return;
            BufferedReader br =
                new BufferedReader(input.getCharacterStream());
            String eventId = "", sessionNo = "", noOfSessions= "",
            line = "";
            while ((line = br.readLine()) != null) {
                if (line.startsWith("eventId"))
                    eventId = line.substring(7);
                else if (line.startsWith("sessionNo"))
                    sessionNo = line.substring(9);
                else if (line.startsWith("noOfSessions"))
                    noOfSessions = line.substring(12);
            }

            Hashtable inputData = new Hashtable();
            inputData.put("eventId", eventId);
            inputData.put("sessionNo",sessionNo);
            inputData.put("noOfSessions", noOfSessions);
            ch.startDocument();
            XMLTools.populateContentHandler(ch, "ESUT",inputData );
            ch.endDocument();
        }
        catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

```
}

public static ByteArrayOutputStream
getInputXML(HttpServletRequest req) {

    HttpSession session = req.getSession(true);
    String noOfSessions =
    (String)session.getAttribute("esut.noOfSessions");
    String sessionNo =
    (String)session.getAttribute("esut.sessionNo");
    String eventId = (String)session.getAttribute("esut.eventId");
    String inputString = "noOfSessions"+noOfSessions+"\n";
    inputString += "sessionNo"+sessionNo+"\n";
    inputString += "eventId"+eventId+"\n";
    String xsltFileName = null;
    ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
    try{
        TransformerFactory transFact =
        TransformerFactory.newInstance();
        if (transFact.getFeature(SAXTransformerFactory.FEATURE)) {
            SAXTransformerFactory saxTransFact =
                (SAXTransformerFactory) transFact;
            TransformerHandler transHand = null;
            if (xsltFileName == null) {
                transHand = saxTransFact.newTransformerHandler();
            } else {
                transHand = saxTransFact.newTransformerHandler(
                    new StreamSource(new File(xsltFileName)));
            }
            transHand.setResult(new StreamResult(baos));
            InputSource input =
            new InputSource(new StringReader(inputString));
            EsutXMLInputCreatorNewSession abc =
```

```
        new EsutXMLInputCreatorNewSession();
        abc.setContentHandler(transHand);
        abc.parse(input);
    } else {
        System.err.println("SAXTransformerFactory is
not supported.");
        System.exit(1);
    }
}
catch (Exception e) {
    e.printStackTrace();
}
return baos;
}
}
```

EsutXMLOutputCreatorDeleteEvent

```
package ca.sfu.elinc.esutfixed;

import java.io.IOException;
import java.io.BufferedReader;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.*;
```

```
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;
import ca.sfu.elinc.esutfixed.util.*;
import ca.sfu.elinc.esutfixed.*;
import java.sql.*;

/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutXMLOutputCreatorDeleteEvent
extends AbstractXMLReader {

    private static final Attributes EMPTY_ATTR = new AttributesImpl();

    public static ByteArrayOutputStream
    getOutputXML(HttpServletRequest req){
        HttpSession session = req.getSession(true);
        String inputString = "";
        String username = (String)session.
        getAttribute("esut.username");
        String role = (String)session.getAttribute("esut.role");
        String eventId = (String)req.getParameter("eventId");
        inputString+="username"+username+"\n";
        inputString+="role"+role+"\n";
        inputString+="eventId"+eventId+"\n";
        String xsltFileName = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
```

```
try{
    TransformerFactory transFact =
    TransformerFactory.newInstance();
    if (transFact.getFeature(SAXTransformerFactory.FEATURE)) {
        SAXTransformerFactory saxTransFact =
            (SAXTransformerFactory) transFact;
        TransformerHandler transHand = null;
        if (xsltFileName == null) {
            transHand = saxTransFact.newTransformerHandler();
        } else {
            transHand = saxTransFact.newTransformerHandler(
                new StreamSource(new File(xsltFileName)));
        }
        transHand.setResult(new StreamResult(baos));
        InputSource input =
        new InputSource(new StringReader(inputString));
        EsutXMLOutputCreatorDeleteEvent abc =
        new EsutXMLOutputCreatorDeleteEvent();
        abc.setContentHandler(transHand);
        abc.parse(input);
    } else {
        System.err.println("SAXTransformerFactory is
        not supported.");
        System.exit(1);
    }
}
catch (TransformerConfigurationException tce){
    System.out.println("TransformerConfigurationException,
    error while transferring the document");
}
catch (TransformerException te){
    System.out.println("TransformerException,
    error while transferring the document");
}
```



```
    }
    catch (Exception e) {
        System.out.println("Some exception");
    }
    return baos;
}

public void parse(InputSource input){
    try{
        ContentHandler ch = getContentHandler();
        if (ch == null) {
            return;
        }
        BufferedReader br =
            new BufferedReader(input.getCharacterStream());
        String username = "",role= "", eventId="", line="";
        Hashtable inputData = new Hashtable();
        while ((line = br.readLine()) != null) {
            if (line.startsWith("username")){
                username = line.substring(8);
                inputData.put("username", username);
            }
            else if (line.startsWith("role")){
                role = line.substring(4);
                inputData.put("role", role);
            }
            else if (line.startsWith("eventId")){
                eventId = line.substring(7);
                inputData.put("eventId", eventId);
            }
        }
    }
}
```

```
        ch.startDocument();
        XMLTools.populateContentHandler(ch,"ESUT", inputData);
        ch.endDocument();
    }
    catch (IOException ioe){
        ioe.printStackTrace();
    }
    catch (SAXException sae)    {
        sae.printStackTrace();
    }

    catch (Exception e) {
        e.printStackTrace();
    }
}
}
```

EsutXMLOutputCreatorNewEvent

```
package ca.sfu.elinc.esutfixed;

import java.io.IOException;
import java.io.BufferedReader;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
```

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;
import ca.sfu.elinc.esutfixed.util.*;
import ca.sfu.elinc.esutfixed.*;
import java.sql.*;

/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutXMLOutputCreatorNewEvent extends AbstractXMLReader {

    private static final Attributes EMPTY_ATTR = new AttributesImpl();

    public static ByteArrayOutputStream
    getSuccessXML(HttpServletRequest req) {
        HttpSession session = req.getSession(true);
        String inputString =
        EsutUtilities.getInputString(req);
        String username = (String)session.
        getAttribute("esut.username");
        String role = (String)session.getAttribute("esut.role");
        String courseId = (String)session.
        getAttribute("esut.courseId");
        inputString+="username"+username+"\n";
        inputString+="role"+role+"\n";
        inputString+="courseId"+courseId+"\n";
    }
}
```

```
String xsltFileName = null;
ByteArrayOutputStream baos =
new ByteArrayOutputStream(1024);
try{
TransformerFactory transFact =
TransformerFactory.newInstance();
if (transFact.getFeature
(SAXTransformerFactory.FEATURE)) {
SAXTransformerFactory saxTransFact =
(SAXTransformerFactory) transFact;
TransformerHandler transHand = null;
if (xsltFileName == null) {
transHand =
saxTransFact.
newTransformerHandler();
} else {
transHand =
saxTransFact.
newTransformerHandler(
new StreamSource
(new File(xsltFileName)));
}
transHand.setResult
(new StreamResult(baos));
InputSource input =
new InputSource
(new StringReader(inputString));
EsutXMLOutputCreatorNewEvent abc =
new EsutXMLOutputCreatorNewEvent();
abc.setContentHandler(transHand);
abc.parse(input);
} else {
System.err.println("SAXTransformer
```

```
Factory is not supported.");
System.exit(1);
}
}
catch (TransformerConfigurationException tce){
System.out.println("TransformerConfiguration-
-Exception while transferring the document");
}
catch (TransformerException te){
System.out.println("TransformerException,
error while transferring the document");
}
catch (Exception e) {
System.out.println("Some exception");
}
return baos;
}

public static ByteArrayOutputStream
getErrorXML(HttpServletRequest req, Vector errors)    {
    HttpSession session = req.getSession(true);
    String inputString = EsutUtilities.
    getInputString(req, errors);
    String xsltFileName = null;
    ByteArrayOutputStream baos =
    new ByteArrayOutputStream(1024);
    try{
        TransformerFactory transFact =
        TransformerFactory.newInstance();
        if (transFact.getFeature(SAXTransformerFactory.FEATURE)){
            SAXTransformerFactory saxTransFact =
                (SAXTransformerFactory) transFact;
            TransformerHandler transHand = null;
```

```
        if (xsltFileName == null) {
            transHand = saxTransFact.newTransformerHandler();
        } else {
            transHand = saxTransFact.newTransformerHandler(
                new StreamSource(new File(xsltFileName)));
        }
        transHand.setResult(new StreamResult(baos));
        InputSource input =
            new InputSource(new StringReader(inputString));
        EsutXMLOutputCreatorNewEvent abc =
            new EsutXMLOutputCreatorNewEvent();
        abc.setContentHandler(transHand);
        abc.parse(input);
    } else {
        System.err.println("SAXTransformerFactory is not
            supported.");
        System.exit(1);
    }
}
catch (TransformerConfigurationException tce){
    System.out.println("TransformerConfigurationException,
        error while transferring the document");
}
catch (TransformerException te){
    System.out.println("TransformerException,
        error while transferring the document");
}
catch (Exception e) {
    System.out.println("Some exception");
}
return baos;
}
```

```
public void parse(InputSource input){
    try{
        ContentHandler ch = getContentHandler();
        if (ch == null) {
            return;
        }
        BufferedReader br =
            new BufferedReader(input.getCharacterStream());
        String username = "",courseId = "",role = "",
            name="",description="",noOfSessions = "",
            maxChoices = "",
            endDate = "", line = "";
        Hashtable inputData = new Hashtable();
        while ((line = br.readLine()) != null) {
            if (line.startsWith("username")){
                username = line.substring(8);
                inputData.put("username", username);
            }
            else if (line.startsWith("courseId")){
                courseId = line.substring(8);
                inputData.put("courseId", courseId);
            }
            else if (line.startsWith("role")){
                role = line.substring(4);
                inputData.put("role", role);
            }
            else if (line.startsWith("name")){
                name = line.substring(4);
                inputData.put("name", name);
            }
            else if (line.startsWith("description")){
                description = line.substring(11);
                inputData.put("description", description);
            }
        }
    }
}
```

```
    }
    else if (line.startsWith("noOfSessions")){
        noOfSessions = line.substring(12);
        inputData.put("noOfSessions", noOfSessions);
    }
    else if (line.startsWith("maxChoices")){
        maxChoices = line.substring(10);
        inputData.put("maxChoices", maxChoices);
    }
    else if (line.startsWith("endDate")){
        endDate = line.substring(7);
        inputData.put("endDate", endDate);
    }
    else if (line.startsWith("errorStart")){
        Vector errors = new Vector();

        String errorLine = br.readLine();
        while (!errorLine.startsWith("errorEnd")) {

            errors.add(errorLine);
            errorLine = br.readLine();
        }
        inputData.put("errors", errors);
    }
}

ch.startDocument();
XMLTools.populateContentHandler(ch,"ESUT", inputData);
ch.endDocument();
}
catch (IOException ioe){
    System.out.println("IOEXCEPTION = "+ioe.getMessage());
    ioe.printStackTrace();
}
```



```
    }
    catch (SAXException sae) {
        System.out.println("SAXEXCEPTION = "+sae.getMessage());
        sae.printStackTrace();
    }

    catch (Exception e) {
        System.out.println("Stack : \n");
        e.printStackTrace();
    }
}

}
```

EsutXMLOutputCreatorNewSession

```
package ca.sfu.elinc.esutfixed;

import java.io.IOException;
import java.io.BufferedReader;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
```

```
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;
import ca.sfu.elinc.esutfixed.util.*;
import ca.sfu.elinc.esutfixed.*;
import java.sql.*;

/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutXMLOutputCreatorNewSession extends AbstractXMLReader{

    private static final Attributes EMPTY_ATTR = new AttributesImpl();

    public static ByteArrayOutputStream
    getSuccessXML(HttpServletRequest req)  {
        HttpSession session = req.getSession(true);
        String inputString = EsutUtilities.getInputString(req);
        String xsltFileName = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
        try{
            TransformerFactory transFact =
            TransformerFactory.newInstance();
            if (transFact.getFeature(SAXTransformerFactory.FEATURE)) {
                SAXTransformerFactory saxTransFact =
                    (SAXTransformerFactory) transFact;
                TransformerHandler transHand = null;
                if (xsltFileName == null) {
```

```
        transHand = saxTransFact.newTransformerHandler();
    } else {
        transHand = saxTransFact.newTransformerHandler(
            new StreamSource(new File(xsltFileName)));
    }
    transHand.setResult(new StreamResult(baos));
    InputSource input =
    new InputSource(new StringReader(inputString));
    EsutXMLOutputCreatorNewSession abc =
    new EsutXMLOutputCreatorNewSession();
    abc.setContentHandler(transHand);
    abc.parse(input);
} else {
    System.err.println("SAXTransformerFactory is
    not supported.");
    System.exit(1);
}
}
catch (TransformerConfigurationException tce){
    System.out.println("TransformerConfigurationException,
    error while transferring the document");
}
catch (TransformerException te){
    System.out.println("TransformerException,
    error while transferring the document");
}
catch (Exception e) {
    System.out.println("Some exception");
}
return baos;
}

public static ByteArrayOutputStream
```

```
getErrorXML(HttpServletRequest req, Vector errors)    {
    HttpSession session = req.getSession(true);
    String inputString =
    EsutUtilities.getInputString(req, errors);
    String noOfSessions =
    (String)session.getAttribute("esut.noOfSessions");
    inputString += "noOfSessions"+noOfSessions+"\n";
    String xsltFileName = null;
    ByteArrayOutputStream baos
    = new ByteArrayOutputStream(1024);
    try{
        TransformerFactory transFact
        = TransformerFactory.newInstance();
        if (transFact.getFeature
        (SAXTransformerFactory.FEATURE)) {
            SAXTransformerFactory saxTransFact =
                (SAXTransformerFactory) transFact;
            TransformerHandler transHand = null;
            if (xsltFileName == null) {
                transHand = saxTransFact.newTransformerHandler();
            } else {
                transHand = saxTransFact.newTransformerHandler(
                    new StreamSource(new File(xsltFileName)));
            }
            transHand.setResult(new StreamResult(baos));
            InputSource input =
            new InputSource(new StringReader(inputString));
            EsutXMLOutputCreatorNewSession abc =
            new EsutXMLOutputCreatorNewSession();
            abc.setContentHandler(transHand);
            abc.parse(input);
        } else {
            System.err.println("SAXTransformerFactory is
```

```
        not supported.");
        System.exit(1);
    }
}
catch (TransformerConfigurationException tce){
    System.out.println("TransformerConfigurationException,
error while transferring the document");
}
catch (TransformerException te){
    System.out.println("TransformerException,
error while transferring the document");
}
catch (Exception e) {
    System.out.println("Some exception");
}
return baos;
}

public void parse(InputSource input){
    try{
        ContentHandler ch = getContentHandler();
        if (ch == null) {
            return;
        }
        BufferedReader br = new BufferedReader
(input.getCharacterStream());
        String username="", eventId = "",role= "", name="",
description="", noOfSeats = "", sessionNo = "", line = "";
        Hashtable inputData = new Hashtable();
        while ((line = br.readLine()) != null) {
            if (line.startsWith("eventId")){
                eventId = line.substring(7);
                inputData.put("eventId", eventId);
            }
        }
    }
}
```

```
    }
    else if (line.startsWith("username")){
        username = line.substring(8);
        inputData.put("username", username);
    }
    else if (line.startsWith("role")){
        role = line.substring(4);
        inputData.put("role", role);
    }
    else if (line.startsWith("name")){
        name = line.substring(4);
        inputData.put("name", name);
    }
    else if (line.startsWith("description")){
        description = line.substring(11);
        inputData.put("description", description);
    }
    else if (line.startsWith("noOfSeats")){
        noOfSeats = line.substring(9);
        inputData.put("noOfSeats", noOfSeats);
    }

    else if (line.startsWith("sessionNo")){
        sessionNo = line.substring(9);
        inputData.put("sessionNo", sessionNo);
    }
    else if (line.startsWith("errorStart")){
        Vector errors = new Vector();

        String errorLine = br.readLine();
        while (!errorLine.startsWith("errorEnd")) {

            errors.add(errorLine);
```

```
                errorLine = br.readLine();
            }
            inputData.put("errors", errors);
        }
    }
    ch.startDocument();
    XMLTools.populateContentHandler(ch,"ESUT", inputData);
    ch.endDocument();
}
catch (IOException ioe){
    System.out.println("IOEXCEPTION = "+ioe.getMessage());
    ioe.printStackTrace();
}
catch (SAXException sae)    {
    System.out.println("SAXEXCEPTION = "+sae.getMessage());
    sae.printStackTrace();
}

catch (Exception e) {
    System.out.println("Stack : \n");
    e.printStackTrace();
}
}

}
```

EsutXMLOutputCreatorPostEvent

```
package ca.sfu.elinc.esutfixed;

import java.io.IOException;
import java.io.BufferedReader;
import java.io.*;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import java.util.*;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;
import ca.sfu.elinc.esutfixed.util.*;
import ca.sfu.elinc.esutfixed.*;
import java.sql.*;

/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */
public class EsutXMLOutputCreatorPostEvent extends AbstractXMLReader {

    private static final Attributes EMPTY_ATTR = new AttributesImpl();

    public static ByteArrayOutputStream
    getOutputXML(HttpServletRequest req) {
        HttpSession session = req.getSession(true);
```



```
String inputString = "";
String username = (String)session.
getAttribute("esut.username");
String role = (String)session.getAttribute("esut.role");
String eventId = (String)req.getParameter("eventId");
inputString+="username"+username+"\n";
inputString+="role"+role+"\n";
inputString+="eventId"+eventId+"\n";
String xsltFileName = null;
ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
try{
    TransformerFactory transFact =
    TransformerFactory.newInstance();
    if (transFact.getFeature(SAXTransformerFactory.FEATURE)) {
        SAXTransformerFactory saxTransFact =
            (SAXTransformerFactory) transFact;
        TransformerHandler transHand = null;
        if (xsltFileName == null) {
            transHand = saxTransFact.newTransformerHandler();
        } else {
            transHand = saxTransFact.newTransformerHandler(
                new StreamSource(new File(xsltFileName)));
        }
        transHand.setResult(new StreamResult(baos));
        InputSource input =
        new InputSource(new StringReader(inputString));
        EsutXMLOutputCreatorPostEvent abc =
        new EsutXMLOutputCreatorPostEvent();
        abc.setContentHandler(transHand);
        abc.parse(input);
    } else {
        System.err.println("SAXTransformerFactory is
        not supported.");
    }
}
```

```
        System.exit(1);
    }
}
catch (TransformerConfigurationException tce){
    System.out.println("TransformerConfigurationException,
error while transferring the document");
}
catch (TransformerException te){
    System.out.println("TransformerException,
error while transferring the document");
}
catch (Exception e) {
    System.out.println("Some exception");
}
return baos;
}
```

```
public void parse(InputSource input){
    try{
        ContentHandler ch = getContentHandler();
        if (ch == null) {
            return;
        }
        BufferedReader br =
            new BufferedReader(input.getCharacterStream());
        String username = "",role= "", eventId="",
            line="";
        Hashtable inputData = new Hashtable();
        while ((line = br.readLine()) != null) {
            if (line.startsWith("username")){
                username = line.substring(8);
                inputData.put("username", username);
            }
        }
    }
}
```

```
        }
        else if (line.startsWith("role")){
            role = line.substring(4);
            inputData.put("role", role);
        }
        else if (line.startsWith("eventId")){
            eventId = line.substring(7);
            inputData.put("eventId", eventId);
        }
    }

    ch.startDocument();
    XMLTools.populateContentHandler(ch,"ESUT", inputData);
    ch.endDocument();
}
catch (IOException ioe){
    System.out.println("IOEXCEPTION = "+ioe.getMessage());
    ioe.printStackTrace();
}
catch (SAXException sae)    {
    System.out.println("SAXEXCEPTION = "+sae.getMessage());
    sae.printStackTrace();
}

catch (Exception e) {
    System.out.println("Stack : \n");
    e.printStackTrace();
}
}

}
```

EsutUtilities

```
package ca.sfu.elinc.esutfixed.util;

import java.io.*;
import org.xml.sax.*;
import org.xml.sax.helpers.AttributesImpl;
import java.util.*;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpSession;

/**
 * @author Baljeet Dhaliwal bsd@sfu.ca
 *
 */
public class EsutUtilities {

    public static String
    getInputString(HttpServletRequest req) {
        Enumeration params = req.getParameterNames();
        String inputString = "";
        while (params.hasMoreElements()){
            String param = (String)params.nextElement();
            inputString+=param+req.getParameter(param)+"\n";
        }
        HttpSession session = req.getSession(true);
        Enumeration attrs = session.getAttributeNames();
        while (attrs.hasMoreElements()){
            String attr = (String)attrs.nextElement();
            if (attr.startsWith("esut.")){
                inputString+=attr.substring(5)+(String)session.

```

```
        getAttribute(attr)+"\n";
    }
}
return inputString;
}

public static String
getInputString(HttpServletRequest req, Vector errors) {
    String inputString = getInputString(req);
    if (errors.size()!=0)
        inputString += "errorStart\n";
    for (int i=0; i <errors.size(); i++){
        inputString += (String)errors.elementAt(i)+"\n";
    }
    if (errors.size()!=0)
        inputString += "errorEnd\n";
    return inputString;
}

public static Vector
validateNewEventParameters(HttpServletRequest req) {
    Vector returnVector = new Vector();
    int noOfSessions =0 , maxChoices=0;
    String param = req.getParameter("name");
    if (isBlank(param) ){
        returnVector.add("Name can not be blank");
    }
    param = req.getParameter("description");
    if (isBlank(param) ){
        returnVector.add("Description can
```

```
        not be blank");
    }
    param = req.getParameter("noOfSessions");
    if (isBlank(param) ){
        returnVector.add("Number of Sessions can
        not be blank");
    }
    else if (!isInteger(param)){
        returnVector.add("Invalid value for
        no of sessions");
    }
    else {
        noOfSessions =
        new Integer(param).intValue();
    }
    param = req.getParameter("maxChoices");
    if (isBlank(param) ){
        returnVector.add("Maximum choices can
        not be blank");
    }
    else if (!isInteger(param)){
        returnVector.add("Invalid value
        for choices");
    }
    else {
        maxChoices = new Integer(param).intValue();
    }
    param = req.getParameter("endDate");
    if (isBlank(param) ){
        returnVector.add("End date can
        not be blank");
    }
}
```

```
    else if (!isDateValid(param)){
        returnVector.add("Invalid date format,
            should be yyyy-mm-dd");
    }
    else if (isEndDatePassed(param)){
        returnVector.add("End date can
            not be in past");
    }
    if (maxChoices>noOfSessions) {
        returnVector.add("No of choices
            can not
            be more than no of sessions");
    }
    return returnVector;
}
```

```
public static Vector
validateNewSessionParameters
(HttpServletRequest req) {
    Vector returnVector = new Vector();
    int noOfSeats =0 ;
    String param = req.getParameter("name");
    if (isBlank(param) ){
        returnVector.add("Name can not be blank");
    }
    param = req.getParameter("description");
    if (isBlank(param) ){
        returnVector.add("Description can
            not be blank");
    }
    param = req.getParameter("noOfSeats");
    if (isBlank(param) ){
```

```
        returnVector.add("Number of Seats  
        can not be blank");  
    }  
    else if (!isInteger(param)){  
        returnVector.add("Invalid value  
        for no of seats");  
    }  
    else {  
        noOfSeats = new Integer(param).intValue();  
    }  
return returnVector;  
}  
  
public static boolean isBlank(String str) {  
    return str.trim().equals("");  
}  
  
public static boolean isInteger(String str) {  
    try {  
        new Integer(str);  
    } catch (NumberFormatException e) {  
        // TODO Auto-generated catch block  
        e.printStackTrace();  
        return false;  
    }  
    return true;  
}  
  
public static boolean isDateValid(String date) {  
    StringTokenizer stk = new StringTokenizer(date, "-");  
    if (stk.countTokens() != 3)  
        return false;  
    String year = stk.nextToken();
```



```
        String month = stk.nextToken();
        String day = stk.nextToken();
        if (!isInteger(year) || !isInteger(month)
            || !isInteger(day))
            return false;
        return true;
    }

    public static boolean isEndDatePassed(String date) {
        StringTokenizer stk =
            new StringTokenizer(date, "-");
        String year = stk.nextToken();
        String month = stk.nextToken();
        String day = stk.nextToken();
        int yr = new Integer(year).intValue();
        int mth = new Integer(month).intValue();
        int dy = new Integer(day).intValue();
        Calendar cal = Calendar.getInstance();
        cal.set(yr, mth-1, dy,0,0,0);
        return new Date().getTime() >
            cal.getTimeInMillis()+86400000L;
    }
}
```

B.1.2 Package ca.sfu.elinc.esutvariable

EsutProcessOutputXML

```
package ca.sfu.elinc.esutvariable;

import java.io.*;
import javax.servlet.http.*;
import javax.xml.parsers.*;
```

```
import org.xml.sax.*;
import org.w3c.dom.*;
import java.util.*;

import java.sql.*;

/**
 * This class is used for processing all the output XML.
 * The methods in this class need to be implemented based on
 * needs of an organization.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */
public class EsutProcessOutputXML {

    /** This method is used for saving a new event on backend
    database. It returns eventId of the new event created.
    Returns -1 if there is an error. The inputXML is the xml for
    the new event, which is based on SaveEvent.xsd schema.
    @param inputXML The XML document in the form of BAOS.
    The xml is based on SaveEvent.xsd schema
    * @param out PrintWriter object used by servlets
    * @return returns the eventId of the event created or -1 in
    case of an error in saving the event.
    */

    public static int processOutputXMLNewEvent
    (ByteArrayOutputStream inputXML, PrintWriter out){
int eventId = -1;
        return eventId;
    }

    /** This method is used for saving a new session on backend
```

```
database. It returns a boolean value representing
successful creation of a session.
 * The inputXML is the xml for the new session, which is
based on SaveSession.xsd schema.
 * @param inputXML The XML document in the form of BAOS.
The xml is based on SaveSession.xsd schema
 * @param out PrintWriter object used by servlets
 * @return returns true if successful in saving the information.
 */

public static boolean
processOutputXMLNewSession
(ByteArrayOutputStream inputXML, PrintWriter out){

    return true;
}

/** This method is used for deleting an event. It returns a
boolean true/false value for successful/failed deletion of
an event. The inputXML is the xml for the event to be
deleted, which is based on DeleteEvent.xsd schema.
 * @param inputXML The XML document in the form of BAOS is
provided. The xml is based on DeleteEvent.xsd schema
 * @param out PrintWriter object used by servlets
 * @return returns true if successful in deleting an event.
 */

public static boolean processOutputXMLDeleteEvent
(ByteArrayOutputStream inputXML, PrintWriter out){

    return true;
}
```

```
/** This method is used for posting an event. It returns a
boolean true/false value for successful/failed in posting
of an event.
 * The inputXML is the xml for the event to be posted, which is
based on PostEvent.xsd schema.
 * @param inputXML The XML document in the form of BAOS is
provided. The xml is based on PostEvent.xsd schema
 * @param out PrintWriter object used by servlets
 * @return returns true if successful in posting an event.
 */

public static boolean processOutputXMLPostEvent
(ByteArrayOutputStream inputXML, PrintWriter out){

    return true;
}

}
```

EsutXMLInputCreatorMain

```
package ca.sfu.elinc.esutvariable;

import java.io.IOException;
import java.io.BufferedReader;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
```

```
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;
import java.util.*;
import ca.sfu.elinc.esutfixed.*;
import java.sql.*;

/**
 * This class is used for creating an input xml for the main page.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */
public class EsutXMLInputCreatorMain {

    /** This method returns an XML BAOS for displaying events.
    The xml is based on ListEvents.xsd
    * @param req The HttpServletRequest object.
    * @return An XML in form of BAOS based on ListEvents.xsd
    */
    public static ByteArrayOutputStream
    getInputXML(HttpServletRequest req) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
        return baos;
    }
}

EsutXMLInputCreatorNewEvent

package ca.sfu.elinc.esutvariable;
```

```
import java.io.*;
import java.sql.*;
import javax.servlet.http.*;
import java.util.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.XMLTools;

/**
 * This class is used for creating an input XML for create event page.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */

public class EsutXMLInputCreatorNewEvent {

    /** This method returns an XML BAOS for displaying
    "create new event". The xml is based on NewEvent.xsd
    * @param req The HttpServletRequest object.
    * @return An XML in form of BAOS based on NewEvent.xsd
    */
    public static ByteArrayOutputStream
    getInputXML(HttpServletRequest req) {
        ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
        return baos;
    }
}
```

B.1.3 Package ca.sfu.elinc.javaxslt

AbstractXMLReader

```
package ca.sfu.elinc.javaxslt;

import java.io.IOException;
import java.util.*;
import org.xml.sax.*;

/**
 * An abstract class that implements the SAX2 XMLReader interface. The
 * intent of this class is to make it easy for subclasses to act as
 * SAX2 XMLReader implementations. This makes it possible, for example,
 * for them to emit SAX2 events that can be fed into an XSLT processor
 * for transformation.
 *
 *
 */
public abstract class AbstractXMLReader
implements org.xml.sax.XMLReader {
    private Map featureMap = new HashMap();
    private Map propertyMap = new HashMap();
    private EntityResolver entityResolver;
    private DTDHandler dtdHandler;
    private ContentHandler contentHandler;
    private ErrorHandler errorHandler;

    /**
     * The only abstract method in this class. Derived classes can
     * parse any source of data and emit SAX2 events to the
     * ContentHandler.
     */
    public abstract void parse(InputSource input) throws IOException,
```

```
        SAXException;

    public boolean getFeature(String name)
        throws SAXNotRecognizedException,
        SAXNotSupportedException {
        Boolean featureValue = (Boolean) this.featureMap.get(name);
        return (featureValue == null) ? false
            : featureValue.booleanValue();
    }

    public void setFeature(String name, boolean value)
        throws SAXNotRecognizedException,
        SAXNotSupportedException {
        this.featureMap.put(name, new Boolean(value));
    }

    public Object getProperty(String name)
        throws SAXNotRecognizedException,
        SAXNotSupportedException {
        return this.propertyMap.get(name);
    }

    public void setProperty(String name, Object value)
        throws SAXNotRecognizedException,
        SAXNotSupportedException {
        this.propertyMap.put(name, value);
    }

    public void setEntityResolver(EntityResolver entityResolver) {
        this.entityResolver = entityResolver;
    }

    public EntityResolver getEntityResolver() {
```



```
        return this.entityResolver;
    }

    public void setDTDHandler(DTDHandler dtdHandler) {
        this.dtdHandler = dtdHandler;
    }

    public DTDHandler getDTDHandler() {
        return this.dtdHandler;
    }

    public void setContentHandler(ContentHandler contentHandler) {
        this.contentHandler = contentHandler;
    }

    public ContentHandler getContentHandler() {
        return this.contentHandler;
    }

    public void setErrorHandler(ErrorHandler errorHandler) {
        this.errorHandler = errorHandler;
    }

    public ErrorHandler getErrorHandler() {
        return this.errorHandler;
    }

    public void parse(String systemId) throws IOException,
        SAXException{
        parse(new InputSource(systemId));
    }
}
```

XMLProcessor

```
package ca.sfu.elinc.javaxslt;

import java.io.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import org.xml.sax.*;
import java.util.*;

/**
 * The class is used for converting an XML document into another
 * XML document with help of an XSLT file.
 * @author <a href="mailto:bsd@sfu.ca">Baljeet Dhaliwal</a>
 *
 */
public class XMLProcessor{

/**
 * Converts an xml stream into another format using
 * an xslt file.
 *
 * @param out a PrintWriter object used
 * for throwing output strings.
 * @param baos the input stream
 * @param xsltFileName the xslt file containing
 * transformation details.
 * @return ByteArrayOutputStream the new formatted
 * document in a stream
 * @see SQLProcessor#changeFormat
 * (PrintWriter, ByteArrayOutputStream, String)
 */
```

```
public ByteArrayOutputStream changeFormat(PrintWriter out,
ByteArrayOutputStream baos, File xsltFileName){

    ByteArrayOutputStream baos1 =
    new ByteArrayOutputStream();
    try{
    ByteArrayInputStream data =
    new ByteArrayInputStream(baos.toByteArray());
    baos1 = new ByteArrayOutputStream(1024);
    TransformerFactory transFact =
    TransformerFactory.newInstance();
    InputStream xmlData = (InputStream)data;
    Source xmlSource = new StreamSource(xmlData);
    Source xsltSource =
    new StreamSource(xsltFileName);
    if (transFact.getFeature
    (SAXTransformerFactory.FEATURE)) {
    SAXTransformerFactory saxTransFact =
    (SAXTransformerFactory) transFact;
    TransformerHandler transHand = null;
    if (xsltFileName == null) {
    transHand =
    saxTransFact.newTransformerHandler();
    } else {
    transHand = saxTransFact.
    newTransformerHandler
    (new StreamSource(xsltFileName));
    }
    Result result =
    new StreamResult(baos1);
    Transformer trans =
    transFact.newTransformer(xsltSource);
    trans.transform(xmlSource, result);
```

```
} else {
    out.println("SAXTransformerFactory is
not supported.");
}
}
catch (TransformerConfigurationException tce){
    out.println("TransformerConfigurationException,
error while transferring the document");
}
catch (TransformerException te){
    out.println("TransformerException,
error while transferring the document");
}
return baos1;
}

/**
 * Converts an xml stream into another format using
an xslt file.
 *
 * @param out a PrintWriter object used for
throwing output strings.
 * @param baos the input stream
 * @param xsltFileName the xslt file containing
transformation details.
 * @return the new formatted document in a stream
 * @see SQLProcessor#changeFormat(PrintWriter,
ByteArrayOutputStream, File)
 */
public ByteArrayOutputStream changeFormat
(PrintWriter out, ByteArrayOutputStream baos,
String xsltFileName){
return changeFormat(out, baos, new File(xsltFileName));
```

```
}
```

```
}
```

XMLTools

```
package ca.sfu.elinc.javaxslt;
```

```
import java.io.*;
```

```
import javax.xml.parsers.DocumentBuilder;
```

```
import javax.xml.parsers.DocumentBuilderFactory;
```

```
import javax.xml.parsers.ParserConfigurationException;
```

```
import org.xml.sax.helpers.DefaultHandler;
```

```
import org.xml.sax.ErrorHandler;
```

```
import org.xml.sax.SAXParseException;
```

```
import java.io.*;
```

```
import org.xml.sax.*;
```

```
import org.xml.sax.helpers.AttributesImpl;
```

```
import java.util.*;
```

```
import javax.servlet.http.HttpServletRequest;
```

```
import java.util.*;
```

```
import org.xml.sax.SAXException;
```

```
import org.w3c.dom.Document;
```

```
import org.w3c.dom.*;
```

```
import java.util.Hashtable;
```

```
/**
```

```
 * @author Baljeet Dhaliwal bsd@sfu.ca
```

```
 *
```

```
 *
```

```
 */
```

```
public class XMLTools {
```

```

public static void populateContentHandler(ContentHandler ch,
String root, Hashtable inputData) throws Exception{
    ch.startElement("", "", root, new AttributesImpl());
    Enumeration en = inputData.keys();
    while (en.hasMoreElements()){
        String tag = (String)en.nextElement();
        Object obj = inputData.get(tag);
        if (obj instanceof Hashtable ){
            populateContentHandler(ch, tag, (Hashtable)obj);
        }
        else if (obj instanceof Vector){
            Vector objs = (Vector)obj;
            ch.startElement("", "", tag, new AttributesImpl());
            System.out.println("Its a vector, tag is "+tag);
            String innerTag = tag.substring(0, tag.length()-1);
            for (int i=0; i<objs.size(); i++){
                Object value = objs.elementAt(i);
                if (value instanceof Hashtable){
                    Hashtable h = (Hashtable)value;
                    populateContentHandler(ch, innerTag, h);
                }
                else if (value instanceof String) {
                    ch.startElement("", "", innerTag,
                    new AttributesImpl());
                    ch.characters(value.toString().toCharArray(),
                    0, value.toString().length());
                    ch.endElement("", "", innerTag);
                }
            }
        }
        ch.endElement("", "", tag);
    }
    else {

```

```
        String value = (String)obj;
        ch.startElement("", "", tag, new AttributesImpl());
        ch.characters(value.toString().toCharArray(),
0, value.toString().length());
        ch.endElement("", "", tag);
    }
}
ch.endElement("", "", root);
}
}
```

B.2 CMS-ESUT

The following packages are used in CMS-ESUT.

B.2.1 Package `ca.sfu.elinc.esutfixed`

This package is used from the basecode shown at B.1.1.

B.2.2 Package `ca.sfu.elinc.esutvariable`

`EsutProcessOutputXML`

```
package ca.sfu.elinc.esutvariable;
```

```
import java.io.*;
import javax.servlet.http.*;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.w3c.dom.*;
import java.util.*;

import java.sql.*;
```

```
/**
 * This class is used for processing all the output XML.
 * The methods in this class are implemented based on SFU CMS.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */
public class EsutProcessOutputXML {

    /** This method is used for saving a new event on backend
    database. It returns eventid of the new event created.
    Returns -1 if there is an error.
    * The inputXML is the xml for the new event, which is
    based on SaveEvent.xsd schema.
    * @param inputXML The XML document in the form of BAOS.
    The xml is based on SaveEvent.xsd schema
    * @param out PrintWriter object used by servlets
    * @return returns the eventId of the event created or -1
    in case of an error in saving the event.
    */

    public static int processOutputXMLNewEvent
    (ByteArrayOutputStream inputXML, PrintWriter out){
int eventId = -1;
        DocumentBuilder parser;
        DocumentBuilderFactory factory =
        DocumentBuilderFactory.newInstance();
        factory.setNamespaceAware(true);
        String username="",courseId="",role="",name="",
        description="",noOfSessions="",maxChoices="",endDate= "";
        try {
            parser = factory.newDocumentBuilder();
            ByteArrayInputStream data =
            new ByteArrayInputStream(inputXML.toByteArray());
            InputStream xmlData = (InputStream)data;
```



```
Document doc = parser.parse(xmlData);
Element root = doc.getDocumentElement();
for (int i = 0; i<root.getChildNodes().getLength();
i++){
    Node tempNode = root.getChildNodes().item(i);
    if (tempNode.getNodeName().equals("username")){
        username = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().
equals("courseId")) {
        courseId = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().equals("role")) {
        role = (tempNode.getFirstChild()).
            getNodeValue();
        if (!isTAOrInstructor(role, out)) {
            out.println("Only TAs and instructors
            are allowed to create an event");
            return -1;
        }
    }
    else if (tempNode.getNodeName().equals("name")) {
        name = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().
equals("description")) {
        description = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().
```

```
        equals("noOfSessions")) {
            noOfSessions = (tempNode.getFirstChild()).
                getNodeValue();
        }
        else if (tempNode.getNodeName().
            equals("maxChoices")){
            maxChoices = (tempNode.getFirstChild()).
                getNodeValue();
        }
        else if (tempNode.getNodeName().
            equals("endDate")){
            endDate = (tempNode.getFirstChild()).
                getNodeValue();
        }
    }
}
/**
    This portion of the code is omitted for its
    sensitive nature. There are 6 lines of code
    here.
    ***/
}
catch (Exception e) {
    e.printStackTrace();
    return -1;
}
return eventId;
}

/** This method is used for saving a new session on backend
database. It returns a boolean value
* representing successful creation of a session.
* The inputXML is the xml for the new session, which is based
on SaveSession.xsd schema.
```

```
* @param inputXML The XML document in the form of BAOS.
The xml is based on SaveSession.xsd schema
* @param out PrintWriter object used by servlets
* @return returns true if successful in saving the information.
*/

public static boolean processOutputXMLNewSession
(ByteArrayOutputStream inputXML, PrintWriter out){
    DocumentBuilder parser;
    DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();
    factory.setNamespaceAware(true);
    String eventId="", role="", name="",description="",
    noOfSeats="", sessionNo="";
    try {
        parser = factory.newDocumentBuilder();
        ByteArrayInputStream data =
        new ByteArrayInputStream(inputXML.toByteArray());
        InputStream xmlData = (InputStream)data;
        Document doc = parser.parse(xmlData);
        Element root = doc.getDocumentElement();
        for (int i = 0; i<root.getChildNodes().getLength();i++){
            Node tempNode = root.getChildNodes().item(i);
            if (tempNode.getNodeName().equals("eventId")) {
                eventId = (tempNode.getFirstChild()).
                getNodeValue();
            }
            else if (tempNode.getNodeName().equals("role")) {
                role = (tempNode.getFirstChild()).getNodeValue();
                if (!isTAOrInstructor(role, out)) {
                    out.println("Only TAs and instructors
                    are allowed to create a session");
                    return false ;
                }
            }
        }
    }
}
```

```
        }
    }
    else if (tempNode.getNodeName().equals("name")) {
        name = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().
        equals("description")){
        description = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().equals("noOfSeats")){
        noOfSeats = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().equals("sessionNo")){
        sessionNo = (tempNode.getFirstChild()).
            getNodeValue();
    }
}

    /**
     * This portion of the code is omitted for its
     * sensitive nature. There are 2 lines of code
     * here.
     */
}

catch (Exception e) {
    e.printStackTrace();
    return false;
}
```

```
        return true;
    }

    /** This method is used for deleting an event. It returns a
    boolean true/false value for successful/failed deletion of
    an event.
    * The inputXML is the xml for the event to be deleted, which
    is based on DeleteEvent.xsd schema.
    * @param inputXML The XML document in the form of BAOS is
    provided. The xml is based on DeleteEvent.xsd schema
    * @param out PrintWriter object used by servlets
    * @return returns true if successful in deleting an event.
    */

    public static boolean processOutputXMLDeleteEvent
    (ByteArrayOutputStream inputXML, PrintWriter out){
        DocumentBuilder parser;
        DocumentBuilderFactory factory =
        DocumentBuilderFactory.newInstance();
        factory.setNamespaceAware(true);
        String eventId="", role="";
        try {
            parser = factory.newDocumentBuilder();
            ByteArrayInputStream data =
            new ByteArrayInputStream(inputXML.toByteArray());
            InputStream xmlData = (InputStream)data;
            Document doc = parser.parse(xmlData);
            Element root = doc.getDocumentElement();
            for (int i = 0; i<root.getChildNodes().getLength(); i++){
                Node tempNode = root.getChildNodes().item(i);
                if (tempNode.getNodeName().equals("eventId")) {
                    eventId = (tempNode.getFirstChild()).
                    getNodeValue();
                }
            }
        }
    }
}
```

```
    }
    else if (tempNode.getNodeName().equals("role")) {
        role = (tempNode.getFirstChild()).getNodeValue();
        if (!isTAOrInstructor(role, out)) {
            out.println("Only TAs and instructors are
                allowed to delete an event");
            return false;
        }
    }
}

/**
    This portion of the code is omitted for its
    sensitive nature. There are 2 lines of code
    here.
    ***/

}
catch (Exception e) {
    e.printStackTrace();
    return false;
}
return true;
}

/** This method is used for posting an event. It returns a
boolean true/false value for successful/failed in posting
of an event.
* The inputXML is the xml for the event to be posted, which is
based on PostEvent.xsd schema.
* @param inputXML The XML document in the form of BAOS is
provided. The xml is based on PostEvent.xsd schema
* @param out PrintWriter object used by servlets
```

```
    * @return returns true if successful in posting an event.
    */

public static boolean processOutputXMLPostEvent
(ByteArrayOutputStream inputXML, PrintWriter out){
    DocumentBuilder parser;
    DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();
    factory.setNamespaceAware(true);
    String eventId="", role="";
    try {
        parser = factory.newDocumentBuilder();
        ByteArrayInputStream data =
        new ByteArrayInputStream(inputXML.toByteArray());
        InputStream xmlData = (InputStream)data;
        Document doc = parser.parse(xmlData);
        Element root = doc.getDocumentElement();
        for (int i = 0; i<root.getChildNodes().getLength();
        i++){
            Node tempNode = root.getChildNodes().item(i);
            if (tempNode.getNodeName().equals("eventId")){
                eventId = (tempNode.getFirstChild()).
                getNodeValue();
            }
            else if (tempNode.getNodeName().equals("role")){
                role = (tempNode.getFirstChild()).
                getNodeValue();
                if (!isTAOrInstructor(role, out)) {
                    out.println("Only TAs and instructors are
                    allowed to delete an event");
                    return false;
                }
            }
        }
    }
}
```

```
    }

    /**
     * This portion of the code is omitted for its
     * sensitive nature. There are 2 lines of code
     * here.
     */
    }

    catch (Exception e) {
        e.printStackTrace();
        return false;
    }

    return true;
}

/** This method checks whether a role is a TA or an instructor.
 * @param role String value for role.
 * @param out PrintWriter object used by servlets.
 * @return returns true if role is TA or instructor.
 */
public static boolean
isTAOrInstructor(String role, PrintWriter out) {
    if (role.equals("TA") || role.equals("instructor")){
        return true;
    }
    return false;
}
}
```

EsutXMLInputCreatorMain

```
package ca.sfu.elinc.esutvariable;
```



```
import java.io.IOException;
import java.io.BufferedReader;
import java.io.*;
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.Statement;
import javax.servlet.*;
import javax.servlet.http.*;
import org.xml.sax.InputSource;
import org.xml.sax.SAXException;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.sqlreader.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.sqlreader.XMLTools;
import java.util.*;
import ca.sfu.elinc.esutfixed.*;
import java.sql.*;

/**
 * This class is used for creating an input XML for the ESUT Main
 * Page of SFU CMS.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */
public class EsutXMLInputCreatorMain extends AbstractXMLReader {

    /** This method parses an InputSource populate the document
    handled by ContentHandler
    * @param input InputSource object, containing information
```

```
    in String format.  
*/  
public void parse(InputSource input){  
    try{  
        ContentHandler ch = getContentHandler();  
        if (ch == null) return;  
        BufferedReader br =  
        new BufferedReader(input.getCharacterStream());  
        String username = "" , courseId = "" , role= "",line ="";  
        while ((line = br.readLine()) != null) {  
            if (line.startsWith("username"))  
                username = line.substring(8);  
            else if (line.startsWith("courseId"))  
                courseId = line.substring(8);  
            else if (line.startsWith("role"))  
                role = line.substring(4);  
        }  
  
    /***  
        This portion of the code is omitted for its  
        sensitive nature. There are 2 lines of code  
        here.  
        ***/  
        Hashtable genDetails = new Hashtable();  
        genDetails.put("username", "bsd");  
        genDetails.put("courseId", "200");  
        genDetails.put("role", "instructor");  
        Vector eventsVector = new Vector();  
        for (int i = 0 ; i <v.size(); i++) {  
            Hashtable event = new Hashtable();  
            Hashtable h = (Hashtable)v.elementAt(i);  
            String eventId = ""+(Integer)h.  
            get("CourseEventID");
```

```

        event.put("eventId", eventId);
        event.put("title", (String)h.get("Name"));
        event.put("description",
            (String)h.get("Description" ) );
        event.put("sessions", ""+
            (Integer)h.get("NoOfSessions" ) );
        event.put("choices", ""+
            (Integer)h.get("MaxChoices"));
        event.put("endDate",
            h.get("Enddate").toString().substring(0,10));
        String posted = ""+(Integer)h.get("EventReleased");
        if (posted !=null && posted.equals("1"))
            event.put("status", "posted");
        else {
            event.put("status", "complete" );
        }
        eventsVector.add( event);
    }
    genDetails.put("events", eventsVector);
    ch.startDocument();
    XMLTools.populateContentHandler(ch,"ESUT",genDetails);
    ch.endDocument();
}
catch (Exception e) {
    System.out.println("Stack : \n");
    e.printStackTrace();
}
}

/** This method returns an XML BAOS for displaying events.
The xml is based on ListEvents.xsd
* @param req The HttpServletRequest object.
* @return An XML in form of BAOS based on ListEvents.xsd

```

```
*/
public static ByteArrayOutputStream
getInputXML(HttpServletRequest req) {
    String username = "bsd",role = "instructor",courseId="200";
    HttpSession session = req.getSession(true);
    Integer userId = (Integer)session.getAttribute("userId");
    if (userId == null) userId = new Integer(102);
    String inputString = "username"+username+"\n";
    inputString += "courseId"+courseId+"\n";
    inputString += "role"+role;
    String xsltFileName = null;
    ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
    try{
        TransformerFactory transFact =
        TransformerFactory.newInstance();
        if (transFact.getFeature(SAXTransformerFactory.FEATURE)){
            SAXTransformerFactory saxTransFact =
                (SAXTransformerFactory) transFact;
            TransformerHandler transHand = null;
            if (xsltFileName == null) {
                transHand = saxTransFact.newTransformerHandler();
            } else {
                transHand = saxTransFact.newTransformerHandler(
                    new StreamSource(new File(xsltFileName)));
            }
            transHand.setResult(new StreamResult(baos));
            InputSource input = new InputSource
            (new StringReader(inputString));
            EsutXMLInputCreatorMain abc =
            new EsutXMLInputCreatorMain();
            abc.setContentHandler(transHand);
            abc.parse(input);
        } else {
```

```
        System.err.println("SAXTransformerFactory is
        not supported.");
        System.exit(1);
    }
}
catch (Exception e) {
    e.printStackTrace();
}
return baos;
}
}
```

EsutXMLInputCreatorNewEvent

```
package ca.sfu.elinc.esutvariable;

import java.io.*;
import java.sql.*;
import javax.servlet.http.*;
import java.util.*;
import org.xml.sax.*;
import org.xml.sax.helpers.*;
import javax.xml.transform.*;
import javax.xml.transform.sax.*;
import javax.xml.transform.stream.*;
import ca.sfu.elinc.javaxslt.sqlreader.AbstractXMLReader;
import ca.sfu.elinc.javaxslt.sqlreader.XMLTools;

/**
 * This class is used for creating an input XML for create event page.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */

public class EsutXMLInputCreatorNewEvent extends AbstractXMLReader {
```

```
/** This method parses an InputSource populate the document
handled by ContentHandler
 * @param input InputSource object, containing information in
String format.
 */
public void parse(InputSource input){
    try{
        ContentHandler ch = getContentHandler();
        if (ch == null) return;
        BufferedReader br =
        new BufferedReader(input.getCharacterStream());
        String username = "", courseId = "", role= "", line ="";
        while ((line = br.readLine()) != null) {
            if (line.startsWith("username"))
                username = line.substring(8);
            else if (line.startsWith("courseId"))
                courseId = line.substring(8);
            else if (line.startsWith("role"))
                role = line.substring(4);
        }
        Hashtable inputData = new Hashtable();
        inputData.put("username", username);
        inputData.put("courseId",courseId);
        inputData.put("role", role);
        ch.startDocument();
        XMLTools.populateContentHandler(ch, "ESUT",inputData );
        ch.endDocument();
    }
    catch (Exception e) {
        System.out.println("Stack : \n");
        e.printStackTrace();
    }
}
```

```
}

/** This method returns an XML BAOS for displaying
"create new event". The xml is based on NewEvent.xsd
 * @param req The HttpServletRequest object.
 * @return An XML in form of BAOS based on NewEvent.xsd
 */
public static ByteArrayOutputStream
getInputXML(HttpServletRequest req) {
    String username = "bsd", role = "instructor",
    courseId = "200";
    HttpSession session = req.getSession(true);

    String inputString = "username"+username+"\n";
    inputString += "courseId"+courseId+"\n";
    inputString += "role"+role;
    String xsltFileName = null;
    ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
    try{
        TransformerFactory transFact =
        TransformerFactory.newInstance();
        if (transFact.getFeature(SAXTransformerFactory.FEATURE)){
            SAXTransformerFactory saxTransFact =
                (SAXTransformerFactory) transFact;
            TransformerHandler transHand = null;
            if (xsltFileName == null) {
                transHand = saxTransFact.newTransformerHandler();
            } else {
                transHand = saxTransFact.newTransformerHandler(
                    new StreamSource(new File(xsltFileName)));
            }
            transHand.setResult(new StreamResult(baos));
            InputSource input =
```

```

        new InputSource(new StringReader(inputString));
        EsutXMLInputCreatorNewEvent abc =
        new EsutXMLInputCreatorNewEvent();
        abc.setContentHandler(transHand);
        abc.parse(input);
    } else {
        System.err.println("SAXTransformerFactory
        is not supported.");
        System.exit(1);
    }
}
catch (Exception e) {
    e.printStackTrace();
}
return baos;
}
}

```

B.2.3 Package `ca.sfu.elinc.javaxslt`

This package is used from the basecode shown at B.1.3.

B.3 Eledge-ESUT

The following packages are used in Eledge-ESUT.

B.3.1 Package `ca.sfu.elinc.esutfixed`

This package is used from the basecode shown at B.1.1.

B.3.2 Package `ca.sfu.elinc.esutvariable`

EsutProcessOutputXML

```
package ca.sfu.elinc.esutvariable;
```



```
import java.io.*;
import javax.servlet.http.*;
import javax.xml.parsers.*;
import org.xml.sax.*;
import org.w3c.dom.*;
import ART1010.Course;
import java.sql.*;

/**
 * This class is used for processing all the output XML.
 * The methods in this class are
 * implemented based on Eledge Learning Management System.
 * @author Baljeet Dhaliwal bsd@sfu.ca
 */
public class EsutProcessOutputXML {

    /** This method is used for saving a new event on backend
    database. It returns eventId of the new event created.
    Returns -1 if there is an error.
    * The inputXML is the xml for the new event, which is based on
    SaveEvent.xsd schema.
    * @param inputXML The XML document in the form of BAOS. The xml
    is based on SaveEvent.xsd schema
    * @param out PrintWriter object used by servlets
    * @return returns the eventId of the event created or -1
    in case of an error in saving the event.
    */

    public static int processOutputXMLNewEvent
    (ByteArrayOutputStream inputXML, PrintWriter out){
    int eventId = -1;
        DocumentBuilder parser;
    DocumentBuilderFactory factory =
```

```
DocumentBuilderFactory.newInstance();
factory.setNamespaceAware(true);
String username="",courseId="",role="",name="",
description="",noOfSessions="",maxChoices="",
endDate= "";
try {
parser = factory.newDocumentBuilder();
ByteArrayInputStream data =
new ByteArrayInputStream(inputXML.
toArray());
InputStream xmlData = (InputStream)data;
Document doc = parser.parse(xmlData);
        toArray());
Element root = doc.getDocumentElement();
for (int i = 0; i<root.getChildNodes().
getLength();i++){
Node tempNode =
root.getChildNodes().item(i);
if (tempNode.getNodeName().
equals("username")) {
username =
(tempNode.getFirstChild()).
getNodeValue();
}
else if (tempNode.getNodeName().
equals("courseId")){
courseId =
(tempNode.
getFirstChild()).
getNodeValue();
}
else if (tempNode.getNodeName().
equals("role")) {
```

```
role = (tempNode.  
getFirstChild()).  
getNodeValue();  
  
        if (!isTAOrInstructor(role, out)) {  
            out.println("Only TAs and instructors are  
allowed to create an event");  
            return -1;  
        }  
    }  
    else if (tempNode.getNodeName().  
equals("name")) {  
        name =  
        (tempNode.getFirstChild()).  
getNodeValue();  
    }  
    else if (tempNode.getNodeName().  
equals("description")) {  
        description =  
        (tempNode.getFirstChild()).  
getNodeValue();  
    }  
    else if (tempNode.getNodeName().  
equals("noOfSessions")) {  
        noOfSessions =  
        (tempNode.getFirstChild()).  
getNodeValue();  
    }  
    else if (tempNode.getNodeName().  
equals("maxChoices")) {  
        maxChoices =  
        (tempNode.  
getFirstChild()).  
getNodeValue();  
    }
```

```
}
else if (tempNode.getNodeName().
equals("endDate")) {
endDate = (tempNode.
getFirstChild()).
getNodeValue();
}
}

Class.forName(Course.jdbcDriver).newInstance();
Connection conn = DriverManager.getConnection
(Course.dbName, Course.
mySQLUser, Course.mySQLPass);
Statement stmt = conn.createStatement();
String updateQuery = "insert into events
(posted, studentIdnumber, title, description,
noofsessions, maxchoices, enddate) values (0,
'" +username+"', '"+name+"', '"+description+
"', '"+noOfSessions+"', '"+
maxChoices+"', '"+endDate+"')";
stmt.executeUpdate(updateQuery, Statement.
RETURN_GENERATED_KEYS );
String sqlQuery = "SELECT LAST_INSERT_ID() from events";
Statement stmt1 = conn.createStatement();
ResultSet rs = stmt1.executeQuery(sqlQuery);
while (rs.next()){
    eventId = rs.getInt(1);
}
}
catch (Exception e) {
e.printStackTrace();
return -1;
}
return eventId;
```

```
}

/** This method is used for saving a new session on backend
database. It returns a boolean value
 * representing successful creation of a session.
 * The inputXML is the xml for the new session, which is based on
SaveSession.xsd schema.
 * @param inputXML The XML document in the form of BAOS.
The xml is based on SaveSession.xsd schema
 * @param out PrintWriter object used by servlets
 * @return returns true if successful in saving the information.
 */

public static boolean processOutputXMLNewSession
(ByteArrayOutputStream inputXML, PrintWriter out){
    DocumentBuilder parser;
    DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();
    factory.setNamespaceAware(true);
    String eventId="", role="", name="",description="",
noOfSeats="", sessionNo="";
    try {
        parser = factory.newDocumentBuilder();
        ByteArrayInputStream data =
        new ByteArrayInputStream(inputXML.toByteArray());
        InputStream xmlData = (InputStream)data;
        Document doc = parser.parse(xmlData);
        Element root = doc.getDocumentElement();
        for (int i = 0; i<root.getChildNodes().getLength(); i++){
            Node tempNode = root.getChildNodes().item(i);
            if (tempNode.getNodeName().equals("eventId")) {
                eventId = (tempNode.getFirstChild()).
                getNodeValue();
            }
        }
    }
}
```

```

    }
    else if (tempNode.getNodeName().equals("role")) {
        role = (tempNode.getFirstChild()).getNodeValue();
        if (!isTAOrInstructor(role, out)) {
            out.println("Only TAs and instructors are
                allowed to create a session");
            return false ;
        }
    }
    else if (tempNode.getNodeName().equals("name")) {
        name = (tempNode.getFirstChild()).getNodeValue();
    }
    else if (tempNode.getNodeName().
equals("description")){
        description = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().equals("noOfSeats")){
        noOfSeats = (tempNode.getFirstChild()).
            getNodeValue();
    }
    else if (tempNode.getNodeName().equals("sessionNo")){
        sessionNo = (tempNode.getFirstChild()).
            getNodeValue();
    }
}
}
Class.forName(Course.jdbcDriver).newInstance();
Connection conn = DriverManager.getConnection
(Course.dbName, Course.mySQLUser, Course.mySQLPass);
Statement stmt = conn.createStatement();
String updateQuery = "insert into sessions (eventid,
title, description, noofseats, sessionno)
values (" + eventId + ", '" + name + "', '" + description + "', " +

```

```
        noOfSeats+", "+ sessionNo+");
        stmt.executeUpdate(updateQuery );
    }
    catch (Exception e) {
        e.printStackTrace();
        return false;
    }
    return true;
}

/** This method is used for deleting an event.
It returns a boolean true/false value
* for successful/failed deletion of an event.
* The inputXML is the xml for the event to be deleted,
which is based on DeleteEvent.xsd schema.
* @param inputXML The XML document in the form of BAOS is
provided. The xml is based on DeleteEvent.xsd schema
* @param out PrintWriter object used by servlets
* @return returns true if successful in deleting an event.
*/
public static boolean processOutputXMLDeleteEvent
(ByteArrayOutputStream inputXML, PrintWriter out){
    DocumentBuilder parser;
    DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();
    factory.setNamespaceAware(true);
    String eventId="", role="";
    try {
        parser = factory.newDocumentBuilder();
        ByteArrayInputStream data =
        new ByteArrayInputStream(inputXML.toByteArray());
        InputStream xmlData = (InputStream)data;
        Document doc = parser.parse(xmlData);
```

```
Element root = doc.getDocumentElement();
for (int i = 0; i<root.getChildNodes().
getLength(); i++){
    Node tempNode = root.getChildNodes().item(i);
    if (tempNode.getNodeName().equals("eventId")) {
        eventId = (tempNode.getFirstChild()).
        getNodeValue();
    }
    else if (tempNode.getNodeName().equals("role")) {
        role = (tempNode.getFirstChild()).getNodeValue();
        if (!isTAOrInstructor(role, out)) {
            out.println("Only TAs and instructors are
            allowed to delete an event");
            return false;
        }
    }
}
Class.forName(Course.jdbcDriver).newInstance();
Connection conn = DriverManager.
getConnection(Course.dbName, Course.mySQLUser,
Course.mySQLPass);
Statement stmt = conn.createStatement();
String updateQuery = "delete from events where
eventId="+eventId;
stmt.executeUpdate(updateQuery );
}
catch (Exception e) {
    e.printStackTrace();
    return false;
}
return true;
}
```



```
/** This method is used for posting an event. It returns a
boolean true/false value
 * for successful/failed in posting of an event.
 * The inputXML is the xml for the event to be posted,
which is based on PostEvent.xsd schema.
 * @param inputXML The XML document in the form of BAOS
is provided. The xml is based on PostEvent.xsd schema
 * @param out PrintWriter object used by servlets
 * @return returns true if successful in posting an event.
 */
public static boolean processOutputXMLPostEvent
(ByteArrayOutputStream inputXML, PrintWriter out){
    DocumentBuilder parser;
    DocumentBuilderFactory factory =
    DocumentBuilderFactory.newInstance();
    factory.setNamespaceAware(true);
    String eventId="", role="";
    try {
        parser = factory.newDocumentBuilder();
        ByteArrayInputStream data =
        new ByteArrayInputStream(inputXML.toByteArray());
        InputStream xmlData = (InputStream)data;
        Document doc = parser.parse(xmlData);
        Element root = doc.getDocumentElement();
        for (int i = 0; i<root.getChildNodes().getLength();i++){
            Node tempNode = root.getChildNodes().item(i);
            if (tempNode.getNodeName().equals("eventId")) {
                eventId = (tempNode.getFirstChild()).
                getNodeValue();
            }
            else if (tempNode.getNodeName().equals("role")) {
                role = (tempNode.getFirstChild()).getNodeValue();
                if (!isTAOrInstructor(role, out)) {
```

```
        out.println("Only TAs and instructors are
        allowed to post an event");
        return false ;
    }
}

Class.forName(course.jdbcDriver).newInstance();
Connection conn = DriverManager.
getConnection(course.dbName, course.mysqlUser,
course.mysqlPass);
Statement stmt = conn.createStatement();
String updateQuery = "update events set posted=1 where
eventId="+eventId;
stmt.executeUpdate(updateQuery );
}
catch (Exception e) {
    e.printStackTrace();
    return false;
}
return true;
}

/** This method checks whether a role is a TA or an instructor.
 * @param role String value for role.
 * @param out PrintWriter object used by servlets.
 * @return returns true if role is TA or instructor.
 */
public static boolean
isTAOrInstructor(String role, PrintWriter out) {
    if (role.equals("TA") || role.equals("instructor")){
        return true;
    }
    return false;
}
```

```
    }  
}
```

EsutXMLInputCreatorMain

```
package ca.sfu.elinc.esutvariable;  
  
import java.io.IOException;  
import java.io.BufferedReader;  
import java.io.*;  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.ResultSet;  
import java.sql.Statement;  
import javax.servlet.*;  
import javax.servlet.http.*;  
import org.xml.sax.InputSource;  
import org.xml.sax.SAXException;  
import org.xml.sax.*;  
import org.xml.sax.helpers.*;  
import javax.xml.transform.*;  
import javax.xml.transform.sax.*;  
import javax.xml.transform.stream.*;  
import ca.sfu.elinc.javaxslt.sqlreader.AbstractXMLReader;  
import ca.sfu.elinc.javaxslt.sqlreader.XMLTools;  
import java.util.*;  
import ca.sfu.elinc.esutfixed.*;  
import java.sql.*;  
import ART1010.Course;  
import ART1010.Student;  
  
/**  
 * This class is used for creating an input XML for the ESUT Main  
 * Page of Eledge. */
```

```
* @author Baljeet Dhaliwal bsd@sfu.ca
*/
public class EsutXMLInputCreatorMain extends AbstractXMLReader {

    /** This method returns an XML BAOS for displaying events.
    The xml is based on ListEvents.xsd
    * @param req The HttpServletRequest object.
    * @return An XML in form of BAOS based on ListEvents.xsd
    */

    public static ByteArrayOutputStream
    getInputXML(HttpServletRequest req) {
        String username = "", role = "student", courseId
        = Course.name;
        HttpSession session = req.getSession(true);
        Student student = (Student)session.getAttribute
        (Course.name + "Student");
        username = student.getIDNumber();
        if (student == null) student = new Student();
        if (student.getIsTA() || student.getIsInstructor() ) {
            role = "instructor";
        }
        String inputString = "username"+username+"\n";
        inputString += "courseId"+courseId+"\n";
        inputString += "role"+role;
        String xsltFileName = null;
        ByteArrayOutputStream baos = new ByteArrayOutputStream(1024);
        try{
            TransformerFactory transFact =
            TransformerFactory.newInstance();
            if (transFact.getFeature(SAXTransformerFactory.FEATURE)){
                SAXTransformerFactory saxTransFact =
                    (SAXTransformerFactory) transFact;
```

```
        TransformerHandler transHand = null;
        if (xsltFileName == null) {
            transHand = saxTransFact.newTransformerHandler();
        } else {
            transHand = saxTransFact.newTransformerHandler(
                new StreamSource(new File(xsltFileName)));
        }
        transHand.setResult(new StreamResult(baos));
        InputSource input =
            new InputSource(new StringReader(inputString));
        EsutXMLInputCreatorMain abc =
            new EsutXMLInputCreatorMain();
        abc.setContentHandler(transHand);
        abc.parse(input);

    } else {
        System.err.println("SAXTransformerFactory is not
            supported.");
        System.exit(1);
    }
}
catch (Exception e) {
    e.printStackTrace();
}
return baos;
}

/** This method parses an InputSource populate
the document handled by ContentHandler
 * @param input InputSource object, containing
information in String format.
 */
public void parse(InputSource input){
```

```
        try{
ContentHandler ch = getContentHandler();
if (ch == null) return;
BufferedReader br =
new BufferedReader(input.getCharacterStream());
String username = "" , courseId = "",
role= "", line ="";
while ((line = br.readLine()) != null) {
if (line.startsWith("username"))
username = line.substring(8);
        else if (line.startsWith("courseId"))
courseId = line.substring(8);
        else if (line.startsWith("role"))
role = line.substring(4);
}

        Class.forName(Course.jdbcDriver).newInstance();
        Connection conn = DriverManager.
getConnection(Course.dbName,
Course.mySQLUser,Course.mySQLPass);
        Statement stmt = conn.createStatement();
        String sqlQuery = "select * from events where
StudentIDNumber='"+username+"' order by eventId";
        ResultSet rs = stmt.executeQuery(sqlQuery);
        Hashtable genDetails = new Hashtable();
        genDetails.put("username", username);
        genDetails.put("courseId", courseId);
        genDetails.put("role", role);
        Vector eventsVector = new Vector();
        while (rs.next()){
            Hashtable event = new Hashtable();
            String eventId = ""+rs.getString("EventId");
            event.put("eventId", eventId);
            event.put("title", rs.getString("Title"));
        }
    }
}
```

```
        event.put("description",
rs.getString("Description"));
String sessions = rs.getString("NoOfSessions");
event.put("sessions",sessions );
event.put("choices", rs.getString("MaxChoices"));
event.put("endDate", rs.getString("EndDate"));
String posted = ""+rs.getInt("Posted");
if (posted !=null && posted.equals("1"))
    event.put("status", "posted");
else {
    Statement stmt1 =
conn.createStatement();
String sqlSession = "select
count(eventid) as Sess from sessions
where eventid =" +eventId;
ResultSet rs1 = stmt1.
executeQuery(sqlSession);
int count =0;
while (rs1.next()){
    count = rs1.getInt(1);
}
if (count==(new Integer(sessions)).
intValue())
event.put("status", "complete");
else
    event.put("status", "incomplete");
}
eventsVector.add( event);
}
genDetails.put("events", eventsVector);
ch.startDocument();
XMLTools.populateContentHandler(ch,"ESUT",genDetails);
ch.endDocument();
```

```
}  
catch (Exception e) {  
    e.printStackTrace();  
}  
}  
}
```

EsutXMLInputCreatorNewEvent

```
package ca.sfu.elinc.esutvariable;  
  
import java.io.*;  
import java.sql.*;  
import javax.servlet.http.*;  
import java.util.*;  
import org.xml.sax.*;  
import org.xml.sax.helpers.*;  
import javax.xml.transform.*;  
import javax.xml.transform.sax.*;  
import javax.xml.transform.stream.*;  
import ca.sfu.elinc.javaxslt.sqlreader.AbstractXMLReader;  
import ca.sfu.elinc.javaxslt.sqlreader.XMLTools;  
import ART1010.Course;  
import ART1010.Student;  
  
/**  
 * This class is used for creating an input XML for create event page.  
 * @author Baljeet Dhaliwal bsd@sfu.ca  
 */  
public class EsutXMLInputCreatorNewEvent extends AbstractXMLReader {  
  
    /** This method parses an InputSource populate  
    the document handled by ContentHandler  
    * @param input InputSource object, containing
```



```
        information in String format.
    */
    public void parse(InputSource input){
        try{
ContentHandler ch = getContentHandler();
if (ch == null) return;
BufferedReader br = new BufferedReader
(input.getCharacterStream());
String username = "", courseId = "",role= "",
line = "";
while ((line = br.readLine()) != null) {
if (line.startsWith("username"))
username = line.substring(8);
        else if (line.startsWith("courseId"))
courseId = line.substring(8);
        else if (line.startsWith("role"))
role = line.substring(4);
}

        Hashtable inputData = new Hashtable();
        inputData.put("username", username);
        inputData.put("courseId",courseId);
        inputData.put("role", role);
        ch.startDocument();
        XMLTools.populateContentHandler(ch, "ESUT",inputData );
        ch.endDocument();
}
catch (Exception e) {
System.out.println("Stack : \n");
e.printStackTrace();
}
}

/** This method returns an XML BAOS for
```

```
    displaying "create new event" events.
    The xml is based on NewEvent.xsd
    * @param req The HttpServletRequest object.
    * @return An XML in form of BAOS based on NewEvent.xsd
    */
public static ByteArrayOutputStream
getInputXML(HttpServletRequest req) {
    String username = "", role = "student",
    courseId = Course.name;
    HttpSession session = req.getSession(true);
    Student student =
(Student)session.getAttribute(Course.name+"Student");
    username = student.getIDNumber();
    if (student == null) student = new Student();
    if (student.getIsTA() ||student.getIsInstructor() ) {
        role = "instructor";
    }
    String inputString = "username"+username+"\n";
    inputString += "courseId"+courseId+"\n";
    inputString += "role"+role;
    String xsltFileName = null;
    ByteArrayOutputStream baos =
new ByteArrayOutputStream(1024);
    try{
    TransformerFactory transFact =
TransformerFactory.newInstance();
    if (transFact.getFeature
(SAXTransformerFactory.FEATURE)) {
    SAXTransformerFactory saxTransFact =
(SAXTransformerFactory) transFact;
    TransformerHandler transHand = null;
    if (xsltFileName == null) {
    transHand = saxTransFact.
```

```
newTransformerHandler();
} else {
transHand = saxTransFact.
newTransformerHandler(
new StreamSource
(new File(xsltFileName)));
}
transHand.setResult(new
StreamResult(baos));
InputSource input =
new InputSource(new
StringReader(inputString));
EsutXMLInputCreatorNewEvent abc =
new EsutXMLInputCreatorNewEvent();
abc.setContentHandler(transHand);
abc.parse(input);
} else {
System.err.println
("SAXTransformerFactory is not
supported.");
System.exit(1);
}
}
catch (Exception e) {
e.printStackTrace();
}
return baos;
}
}
```

B.3.3 Package ca.sfu.elinc.javaxslt

This package is used from the basecode shown at B.1.3.

Appendix C

Ant Build File and CSS

C.1 build.xml

```
<?xml version="1.0"?>

<!--
    Build file for the EventSignup Project.
    Baljeet Dhaliwal
    11/2004
-->

<project name="esut" default="build" basedir=".">
    <!-- Configure the context path for this application -->
    <property file="build.properties"/>
    <property name="tomcat.home" value="c:/Program Files/Apache
Software Foundation/Tomcat 5.5" />
    <property name="project.name" value="AT0-esut" />
    <property name="build.dir" value="${basedir}/build" />
    <property name="src.dir" value="${basedir}" />
    <property name="src.java.dir" value="${src.dir}" />
    <property name="docs.dir" value="${basedir}/docs" />
    <property name="docs.api.dir" value="${docs.dir}/api" />
    <property name="web.dir" value="${basedir}/web" />

```

```

<property name="lib.dir" value="${basedir}/lib" />
<property name="dist.dir" value="${basedir}/dist" />
<property name="deploy.dir" value="${tomcat.home}/webapps" />
<path id="classpath">
  <fileset dir="${lib.dir}">
    <include name="**/*.jar"/>
  </fileset>
</path>
<target name="clean" >
  <delete dir="${build.dir}" />
</target>
<target name="prepare" depends="clean"
  description="Create build directories.">
  <mkdir dir="${build.dir}/${project.name}" />
  <mkdir dir="${build.dir}/${project.name}/WEB-INF" />
  <mkdir dir="${build.dir}/
  ${project.name}/WEB-INF/classes"/>
  <mkdir dir="${build.dir}/${project.name}/WEB-INF/lib"/>
</target>
<target name="deploy.copyJars">
<copy todir="${build.dir}/${project.name}/WEB-INF/lib">
  <fileset dir="lib">
    <include name="**/*.jar"/>
    <include name="**/*.dtd"/>
  </fileset>
</copy>
</target>
<!-- Executable Targets -->
<target name="build" depends="prepare"
  description="Compile Java files and copy static files." >
  <javac srcdir="${src.java.dir}"
  destdir="${build.dir}/${project.name}/WEB-INF/classes"
  deprecation="on">

```

```

<include name="**/*.java" />
  <classpath refid="classpath"/>
</javac>
<copy todir="${build.dir}/${project.name}/images"
preserverlastmodified="true">
  <fileset dir="${basedir}/images" />
</copy>
<copy todir="${build.dir}/${project.name}/xslt"
preserverlastmodified="true">
  <fileset dir="${basedir}/xslt" />
</copy>
  <copy todir="${build.dir}/${project.name}/WEB-INF"
preserverlastmodified="true">
  <fileset dir="${basedir}/WEB-INF" />
</copy>
  <copy todir="${deploy.dir}" preserverlastmodified=
"true">
  <fileset dir="${build.dir}"></fileset>
</copy>
</target>
<target name="build.war" depends="build">
<delete dir="${dist.dir}" />
<mkdir dir="${dist.dir}" />
<!-- create a war file for distribution -->
<jar destfile="${dist.dir}/${project.name}.war"
basedir="${build.dir}/${project.name}"/>
<delete dir="${build.dir}" />
</target>
<target name="deploy.war" depends="build.war" >
<delete file="${deploy.dir}/${project.name}.war" />
  <copy todir="${deploy.dir}"
preserverlastmodified="true">
<fileset dir="${dist.dir}">

```

```
<include name="*.war"/>
</fileset>
  </copy>
</target>
  <target name="redeploy.war" depends=" deploy.war"
    description="redeploys the war file in tomcat">
</target>
</project>
```

C.2 stylesheet.css

```
body {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
background-color: white;
font-size: 100%;
color: #555;
}

a {text-decoration: font-size: 60%;}

#container {
margin: 15px;
background-color: #D6DBBF;
}
#content {
padding: 15px;
background-color: #EBF0D1;
margin: 20px;
width: 97%;
}
#banner {
height: 74px;
```

```
padding: 10px;
margin: 15px;
background: url
(http://www.sfu.ca/~jtoal/images/esut/SFU_logo.gif);
background-repeat: no-repeat;
background-color: #D6DBBF;
}
.bannertext {
margin-left: 85px;
color: #5B8800;
font-style: normal;
font-size: 200%;
vertical-align: bottom;
}
#footerText {
clear: both;
padding: 5px;
font-size: 75%;
text-align: right;
margin-right: 20px;
}
tr.live{
background-color: white;
border-style: 1px black;
}
th{
text-align: left; color: #555;
border-color: #000000;
}
td{
text-align: left;
border-color: #000000;
}
```



```
td.eventtext {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 80%;
vertical-align:middle;
}

td.eventlive {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 80%;
font-style: bold;
vertical-align:middle;
color: black;
}

td.eventtitle {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 90%;
width:250;
vertical-align:top;
border:1px solid #D6DBBF;
font-style:bold;
}

td.othertitle {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 90%;
vertical-align:top;
border:1px solid #D6DBBF;
font-style:bold;
}

th.eventtitle {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
```

```
lucida, arial, helvetica, sans-serif;
font-size: 90%;
font-weight:normal;
width:250;
vertical-align:top;
border:1px solid #D6DBBF;
font-style:bold;
}
th.othertitle {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 90%;
font-weight:normal;
vertical-align:top;
border:1px solid #D6DBBF;
font-style:bold;
}
th.blank {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 90%;
font-weight:bold;
vertical-align:top;
border:1px solid #000000;
font-style:bold;
}
td.eventinput {
font-family: "Trebuchet MS", "Bitstream Vera Sans", verdana,
lucida, arial, helvetica, sans-serif;
font-size: 90%;
vertical-align:top;
border:1px solid #D6DBBF;
text-align: right;
```

```
font-style:bold;
}
td.noDeco{
border-style:none;
}
.headertext {
color: #333;
font-style: bold;
font-size: 110%;
}
.footerText {
font-size:100%;
text-decoration:none;
color: #E90649
}
.highLightText2 {
color: #0099CC
}
.errorMessage {
color: #E90649
}
.roundcont {
width: 100%;
background-color: #D6DBBF;
color: #fff;
}
.roundcont p {
margin: 0 10px;
}
.roundtop {
background: url(http://www.sfu.ca/~jtoal/images/esut/tr.gif)
no-repeat top right;
}
```

```
.roundbottom {  
background: url(http://www.sfu.ca/~jtoal/images/esut/br.gif)  
no-repeat top right;  
}  
img.corner {  
    width: 15px;  
    height: 15px;  
    border: none;  
    display: block !important;  
}  
<!-- end rounded corner code-->
```

Bibliography

- [1] Ant Apache. <http://ant.apache.org/>.
- [2] APICS. *APICS Dictionary*. APICS, 8th edition, 1995.
- [3] Charles Ashbacher. *XML in 24 hours*. Techmedia, New Delhi, first edition, 2000.
- [4] Sami Asiri. Open source software. *SIGCAS Comput. Soc.*, 33(1):2, 2003.
- [5] Judith Barnard. A new reusability metric for object-oriented software. *Software Quality Control*, 7(1):35–50, 1998.
- [6] Robert Biddle and Ewan Tempero. Explaining issues in software reusability via an audio system analogy. In *ACSE '97: Proceedings of the 2nd Australasian conference on Computer science education*, pages 71–77, New York, NY, USA, 1996. ACM Press.
- [7] Building Blocks. <http://www.blackboard.com/extend/dev/>.
- [8] Bray J. Brockbank. *The AMA Handbook of E-learning*. AMACOM - American Management Association, 2003.
- [9] Eric M Burke. *Java and XSLT*. Oreilly, 2001.
- [10] Cancore. <http://www.cancore.ca/>.
- [11] Abhijeet Chavan and Shireen Pavri. Open-source learning management with moodle. *Linux J.*, 2004(128):2, 2004.
- [12] Learning Circuits. <http://www.learningcircuits.org/2005/jun2005/>.
- [13] Geoff Collier and Robby Robson. What is the open knowledge initiative?
- [14] Robert A. Comerford. *Strategic Management, 2nd Edition*.
- [15] IMS Global Learning Consortium. <http://www.imsglobal.org/>.
- [16] World Wide Web Consortium. <http://www.w3.org/>.
- [17] Craig Counterman, Glenn Golden, Rachel Gollub, Mark Norton, Charles Severance, and Lance Speelmon. Technical report sakai project - sakai java framework.

- [18] Rabelani Dagada and Maria Jakovljevic. 'where have all the trainers gone?' e-learning strategies and tools in the corporate training environment. In *SAICSIT '04: Proceedings of the 2004 annual research conference of the South African institute of computer scientists and information technologists on IT research in developing countries*, pages 194-203, , Republic of South Africa, 2004. South African Institute for Computer Scientists and Information Technologists.
- [19] David Herron David Garmus. *Function Point Analysis, Measurement Practices for Successful Software Projects*. Addison-Wesley, 2001.
- [20] Information Economy defined by Australian Government Information Office. <http://www.agimo.gov.au/publications/2001/11/ar00-01/glossary>.
- [21] LMS defined by Brandon Hall. <http://www.brandon-hall.com/>.
- [22] Industrial Revolution defined by Professor Gerhard Rempel. <http://mars.acnet.wnec.edu/grempel/courses/wc2/lectures/industrialrev.html>.
- [23] SAKAI defined by SAKAI website. <http://www.sakaiproject.org/>.
- [24] JAXP defined by Sun Microsystems website. <http://java.sun.com/webservices/jaxp/docs.html>.
- [25] Open Services Interface Definition. <http://www.okiproject.org/specs/>.
- [26] EduTools. <http://www.edutools.info>.
- [27] Bill Brykczynski et al. *Software Engineering Project Management*. IEEE Computer Society, 2000.
- [28] Hoffman Fogarty, Blackstone. *Production and Inventory Management, 2nd edition*. South Western Publishing Company.
- [29] William Frakes and Carol Terry. Software reuse: metrics and models. *ACM Comput. Surv.*, 28(2):415-435, 1996.
- [30] Benjamin Hansen. Discussion of order specification decoupling line.
- [31] Robert Heckel. *The Greatest Summer Job in the Whole, Wide World*. Silk Label Books, 2001.
- [32] Pew Internet and American Life Project (2001). The internet and education. pew internet and american life project.
- [33] Investopedia. Inventory turnover ratio - <http://www.investopedia.com/university/ratios/inventoryturnover.asp>.

- [34] Sushant Jain, Ratul Mahajan, and Dan Suci. Translating xslt programs to efficient sql queries. In *WWW '02: Proceedings of the 11th international conference on World Wide Web*, pages 616–626, New York, NY, USA, 2002. ACM Press.
- [35] Jeffrey Kahn. A developer’s guide to the repository open service interface definition.
- [36] Kyong-Jee Kim, Curtis J. Bonk, and TingTing Zeng. Surveying the future of workplace e-learning: the rise of blending, interactivity, and authentic learning. *eLearn*, 2005(6):2, 2005.
- [37] Shinji Kusumoto, Masahiro Imagawa, Katsuro Inoue, Shuuma Morimoto, Kouji Matsusita, and Michio Tsuda. Function point measurement from java programs. In *ICSE '02: Proceedings of the 24th International Conference on Software Engineering*, pages 576–582, New York, NY, USA, 2002. ACM Press.
- [38] Advanced Distributed Learning. <http://www.adlnet.org/scorm/index.cfm>.
- [39] Zhaolin Li and Susan H. Xu. Managing a single product assemble-to-order system with technology innovations.
- [40] GNU General Public License. <http://www.gnu.org/copyleft/gpl.html>.
- [41] David Longstreet. *Function Points Analysis Training Course*. Software Metrics, October 2004.
- [42] Mitchell D. Lubars. Affording higher reliability through software reusability. *SIGSOFT Softw. Eng. Notes*, 11(5):39–42, 1986.
- [43] Moodle. <http://www.moodle.org>.
- [44] Don Morrison. *E-Learning Strategies*. John Wiley & Sons Ltd, 2003.
- [45] Debbie Murray. E-learning for the workplace - creating canadas lifelong learners. 2001.
- [46] Shari Lawrence Pfleeger Norman E Fenton. *Software Metrics*. PWS Publishing Company, 1997.
- [47] University of Utah. Eledge at <http://eledge.sourceforge.net/>.
- [48] OKI. <http://www.okiproject.org/>.
- [49] Morten F Paulsen. *Online Education - Learning Management Systems*. NKI Forlaget, 2003.
- [50] WebCT Powerlinks. <http://www.webct.com/powerlinks/>.
- [51] John F Proud. *Master Production Schedule, 2nd Edition*. John Wiley & Sons.
- [52] Chetan Soman. Make-to-order and make-to-stock in food processing industries.

- [53] Sourceforge. <http://sourceforge.net/>.
- [54] Bedir Tekinerdogan. *Synthesis-Based Software Architecture Design*. PhD thesis, University of Twente, Enschede, The Netherlands., 2000.
- [55] Scott Thorne, Chuck Shubert, and Jeff Merriman. *oki architecture overivew*.
- [56] J. Voas, J. Payne, R. Mills, and J. McManus. Software testability: an experiment in measuring simulation reusability. In *SSR '95: Proceedings of the 1995 Symposium on Software reusability*, pages 247–255, New York, NY, USA, 1995. ACM Press.
- [57] Whybark Vollman, Berry. *Manufacturing Planning and Control Systems, 4th edition*. McGraw-Hill.
- [58] Zhiping Wang and Birger Rapp. Modelling customization in production inventory system.
- [59] Peter Wegner. Capital-intensive software technology. *IEEE Software*, 1,3:7–45, 1984.
- [60] XML. <http://www.w3.org/xml/>.
- [61] XSLT. <http://www.w3.org/tr/xslt>.
- [62] W. Yin, M. M. Tanik, D. Y. Y. Yun, T. J. Lee, and A. G. Dale. Software reusability: a survey and a reusability experiment. In *ACM '87: Proceedings of the 1987 Fall Joint Computer Conference on Exploring technology: today and tomorrow*, pages 65–72, Los Alamitos, CA, USA, 1987. IEEE Computer Society Press.