



THE UNIVERSITY OF QUEENSLAND

Technical Conference and Meetings Management System

Tim Dettrick

The School of Information Technology
and Electrical Engineering

The University of Queensland

Submitted for the degree of
Bachelor of Engineering (Honours)
in the division of
Software Engineering

October 27, 2004

October 27, 2004

Head of School
School of Information Technology and Electrical Engineering
The University of Queensland
St Lucia QLD 4072

Dear Professor Bailes,

In accordance with the requirements of the degree of “Bachelor of Engineering (Honours)” in the School of Information Technology and Electrical Engineering, I submit the following thesis, entitled:

“Technical Conference and Meetings Management System”

This thesis was performed under the supervision of Associate Professor Brian Lovell. I declare that the work submitted in this thesis is my own, except as acknowledged in the text and footnotes, and has not been previously submitted for a degree at the University of Queensland or any other institution.

Yours Sincerely,

Tim Dettrick

Abstract

COnference and **M**eeting **M**anagement with **E**lectronic **N**otification and **C**ommittee **E**xchanges (COMMENCE) is a web-based conference management system designed to facilitate the running of technical conferences. It aims to provide an easy-to-use conference management tool with flexible hosting requirements under an open source license.

The scope of the COMMENCE system has been expanded from its initial focus on academic paper submission and review. New features include Post-Submission Management (PSM), a conference registration framework and improved customization. PSM provides functionality for planning the conference itself, including automatic and manual session slot allocation for papers and the generation of a conference proceedings CD. The conference registration framework is customizable through XML. Conferences can customize their registration form to meet their individual needs, produce registration statements, verify registration payment and track registration statistics.

Contents

1	Introduction	1
1.1	COMMENCE	1
1.2	Focus of Thesis Project	2
2	Background	3
2.1	An Overview of the COMMENCE system structure	3
2.2	Licensing	4
2.2.1	Open Source	4
2.2.2	Attracting Developers	4
2.2.3	The Mozilla Public License	5
2.3	Technologies & Processes	6
2.3.1	PHP	6
2.3.2	MySQL	6
2.3.3	JavaScript	6
2.3.4	Refactoring	6
2.4	Existing Conference Management Systems	7
2.4.1	START (Version 2)	7
2.4.2	SciCon ^{PHP}	7
2.4.3	WCMT	9
2.4.4	OpenConf	10
3	Planning for Expansion	11
3.1	Expanding, Extending and Improving COMMENCE	11
4	Implementing the Plan	15
4.1	Improving the Overall Design	15
4.1.1	Restructuring the Settings Framework	15
4.1.2	Refactoring the User Registration Form	16
4.1.3	Standardizing the Date Format	16
4.1.4	Improving Customization and Localization	17
4.1.5	Database access with ADOdb	17
4.1.6	Migrating to UTF-8	19
4.1.7	Improving HTML code	19
4.2	Post-Submission Management	20

4.2.1	Overview	20
4.2.2	Rooms and Presentation Types	20
4.2.3	Sessions	21
4.2.4	Modifying the Accept/Reject Paper page and Letter	21
4.2.5	Automatically Scheduling Sessions	22
4.2.6	Rescheduling Slots Manually	22
4.2.7	Conference Program	23
4.2.8	Generating the Conference CD structure	23
4.3	Attendee Registration	25
4.3.1	Overview	25
4.3.2	XML and XSLT	25
4.3.3	The Registration Form XML Definition	25
	Payment times	26
	Registration types	26
	Memberships	27
	Choices	27
4.3.4	Conference Registration Statement	28
4.3.5	Verifying Conference Registration Forms	28
4.3.6	Registration Statistics	29
4.3.7	Changing Registration Form Details	29
5	Results of Work	31
5.1	Post-Submission Management	31
5.2	Conference Registration	32
5.3	Customization	33
5.4	ADODB	33
5.5	Documentation	34
5.6	Refactoring	35
6	Future Development	37
6.1	PHP5 Compatibility	37
6.2	Letter Composition	37
6.3	Database Transactions	37
6.4	Increased registration form flexibility	38
6.5	Using XML more broadly in COMMENCE	38
7	Conclusion	41
A	PSM Database Definitions	45
B	Session Rescheduling JavaScript	47
C	Example Conference Program	51
D	Registration Form DTD	53

E XML Registration Form Specification	55
F Example Conference Registration Form Definition	61
G Example Conference Registration Statement	69
H Example Conference Registration Statistics	71

List of Figures

2.1	COMMENCE Overview	3
2.2	Screenshot of SciCon ^{PHP}	8
2.3	Screenshot of WCMT	9
2.4	Screenshot of OpenConf	10
4.1	Original Settings Table	15
4.2	New Settings Table	16
4.3	JavaScript calendar and date input fields	17
4.4	New COMMENCE Settings page	18
4.5	Relationships between entities in PSM	20
4.6	The Add Session page	21
4.7	Manually rearranging session slots in COMMENCE	22
4.8	Building the CD Structure in COMMENCE	23
4.9	Compiling and Downloading the CD Structure in COMMENCE	24
4.10	Payment times in a COMMENCE conference registration form	26
4.11	Registration types in a COMMENCE conference registration form	26
4.12	Memberships in a COMMENCE conference registration form	27
4.13	A choice with options in a COMMENCE conference registration form	27
4.14	A choice with a quantity in a COMMENCE conference registration form	28
4.15	Verifying the payment of a registration form in COMMENCE	28
4.16	Changing the registration settings in COMMENCE	29
C.1	Generated conference program in COMMENCE	51
G.1	Conference registration statement in COMMENCE	69
H.1	Conference registration statistics in COMMENCE	71

Chapter 1

Introduction

Conference management software is an expanding area of software development, as conference conveners take advantage of the increased availability of access to the world wide web. Manual organization of the submitting and reviewing of papers is generally a long a tedious process, with ample opportunity for loosing information or making mistakes. Automating conference preparation organizes and vastly reduces the workload for the conference convener, while also making the process more accessible to all those involved.

1.1 COMMENCE

Conference and **M**eeting **M**anagement with **E**lectronic **N**otification and **C**ommittee **E**xchanges (COMMENCE) is a conference management system designed to facilitate the running of technical conferences. It is written in PHP and uses a MySQL database as a back-end. Its initial focus was on electronic paper submission and reviewing.

At the beginning of 2004, COMMENCE contained the following features:

- Paper submission
 - Papers are able to be submitted in electronic form to a central location for review and possible acceptance.
- Allocation of tracks, keywords, and abstracts to papers
 - Papers are allocated a track (essentially a main topic), keywords (topic areas to which the paper is related) and descriptive abstract inside the system.
- Bidding for paper reviews
 - Bids are taken from reviewers for individual paper reviews. Final allocations are made by the convenor.
- Collection of reviewer ratings and comments
- Automated mass email notifications
 - The system automatically notifies users of submission stage changes and other major events.

The vision for the eventual form of COMMENCE, as conceived by project founder Brian Lovell, was the creation of a web-based system capable of assisting the convenor in all stages of running a conference, with the intention being able to run a conference with over 1000 participants by 2008. While the existing features of COMMENCE went some way towards this goal, a significant amount of work was and still is required to meet this goal.

COMMENCE is also an open-source project, and is intended for use by anybody wishing to run a technical conference. As such, it requires additional flexibility to handle the wide range of platforms and conference requirements with which it must work. There are also financial imperatives in maintaining COMMENCE as a flexible system, in the form of a support contract with the **SouthEast SAS Users Group** (SESUG), and development imperatives which will be discussed later in this document.

1.2 Focus of Thesis Project

The goal of this thesis project was to extend COMMENCE to handle other parts of conference organisation, including the scheduling of paper presentations and attendee registration. In addition, COMMENCE required some internal improvements to lay the groundwork for future development initiatives.

Chapter 2

Background

2.1 An Overview of the COMMENCE system structure

COMMENCE is composed of two separate components: the web server hosting its code, and the relational database storing its conference information. The advantage of this architecture is that the database which stores the conference data does not have to be located on the same machine as the web server hosting the conference. An organisation can use their existing web-hosting to host a conference running COMMENCE, and link it to a separate database server. This allows COMMENCE to take advantage of backup procedures for existing servers, rather than force conveners to locate all the servers on a new machine.

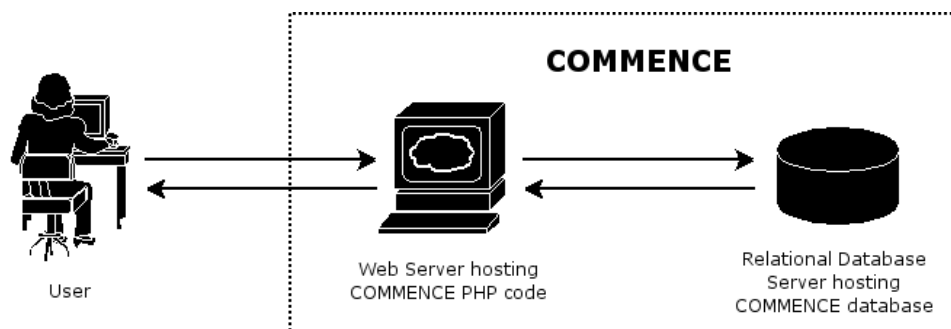


Figure 2.1: COMMENCE Overview

2.2 Licensing

2.2.1 Open Source

COMMENCE is open-source software, with all its code licensed under the **Mozilla Public License (MPL)**. Open source is no longer a new occurrence in the realm of software development, and has been popularised by products such as the Linux operating system, the Apache HTTP Server and more recently the Mozilla web browsers. The Open Source Initiative, a non-profit corporation dedicated to promoting the concept of open source, perhaps best states driving force behind open source:

The **basic idea behind open source** is very simple: When programmers can read, redistribute, and modify the source code for a piece of software, the software evolves. People improve it, people adapt it, people fix bugs. And this can happen at a speed that, if one is used to the slow pace of conventional software development, seems astonishing.

(Open Source Initiative 2004)

The reason why COMMENCE is open-source software is based on this concept. COMMENCE was not started with the intention of drawing a profit from its development. Its focus was instead to produce a piece of software for use in running conferences, with the benefits from using that software being the main reward for its development. Thus the most important thing for COMMENCE is the evolution of its code in a direction benefiting those who have contributed to it. With no reason not to license COMMENCE as open source software, and with the developmental benefits provided by it, COMMENCE was listed in the SourceForge online software repository and published under an open source license.

2.2.2 Attracting Developers

However, the benefits of an open source license do not come automatically, and there are a number of factors which contribute to their impact. Studies have shown that a higher rate of development on a project tends to increase developer interest and contributions to an open source project (Crowston & Scozzi 2002). Additionally, studies also suggest that because open source developers are driven by self interest, the number of developers interested in contributing is related to the number of developers who face the problem the project aims at fixing (Crowston & Scozzi 2002). There is also evidence that the availability of skills is a key factor in the success of open source projects (Crowston & Scozzi 2002).

In relation to the first point, COMMENCE can best attempt to acquire new developers, and thus further its own goals, by trying to deliver new features regularly, and avoid long periods of time without updates. With the consideration of other projects like COMMENCE, this could be seen as a competition between competing projects to get the most interesting features in the shortest time-frame, in an effort to attract developers to the most promising project.

On the second of the three points, the mere fact that COMMENCE is not the only project of its kind suggests that there is at least some limited interest in solving the problem of conference management. However the absolute size of that interest is not as great as for more general

projects. Additionally, the focus of the project will also determine the level of interest in it. For instance, if COMMENCE were to become tied to uncommon or obsolete technologies, the number of people that could take advantage of it would diminish, and developer interest could likewise be expected to diminish. Thus flexibility in design and use is key to the success of COMMENCE in the long term.

In terms of developer skills, COMMENCE mainly relies on PHP programming and writing of SQL queries. Neither are particularly rare or hard to acquire, however just as uncommon technologies can prevent others using COMMENCE, so too can they reduce the number of developers with the required skills.

2.2.3 The Mozilla Public License

COMMENCE is currently licensed under the MPL, which is a less common yet not obscure open source license. The license allows free use of the code under a number of conditions. Two of the main requirements are that any modifications to files containing the original code itself must also be licensed under the MPL, and that any contributor to the code waives any rights to enforce patents on their contributions (Behlendorf 1999). This is similar to the more commonly known GPL, however in the case of the MPL the requirement to license changes under the same license only applies to files part of the original work. Thus, for instance, developing a plug-in for an MPL application under another license is acceptable, providing changes to the original files that allow the plug-in to work are licensed under the MPL.

The MPL license allows COMMENCE to grow and expand as an open source project and benefit from changes made by the community. Under the same license however, it also allows the development of distinct new sections which are not covered by an open source license. While this feature of the license is not currently in use, it may encourage future development.

2.3 Technologies & Processes

2.3.1 PHP

PHP (**PHP: Hypertext Preprocessor**, a recursive acronym) is scripting language focused on web development applications. Web pages contain sections of PHP code, which are processed and evaluated to form a normal web page. PHP is a server-side language, meaning that the code is processed before being sent to the client computer. PHP can be run on both UNIX-like platforms - such as Linux, Solaris and BSD - and Microsoft Windows server platforms. PHP is licensed under its own PHP License, currently at version 3.0, which is open source in nature.

COMMENCE is mostly written in PHP4, which is the current stable version of PHP. However, PHP5 has recently emerged from beta, so COMMENCE will probably need to be made compatible with the new version in the near future.

2.3.2 MySQL

MySQL is a open-source database server, which runs on both UNIX-like and Microsoft Windows platforms. MySQL offers a solid choice to open source developers looking for a free, general-purpose database server to use with their software.

COMMENCE utilizes MySQL for its data storage, which includes all submitted papers, user accounts, and settings designed to be change by the administrator after installation.

2.3.3 JavaScript

JavaScript is a client-side scripting language used in web page design. It is commonly used for dynamic content in web pages where fast responsiveness to the user is required, as unlike server-side languages like PHP, all the processing is done on the client computer.

Prior to this thesis project, COMMENCE utilized JavaScript mainly for the administrator interface menus, however the advantages of client-side processing leant to utilizing it in further applications.

2.3.4 Refactoring

Refactoring is the process of changing a software system in such a way that it does not alter the external behaviour of the code yet improves its internal structure.

(Fowler 1999)

The concept of refactoring is reduce the complexity of code in order to make it more comprehensible and maintainable. A good analogy is that of moving from a prototype to a production model in traditional engineering. The prototype will often have room for components that were never incorporated, features that are superfluous to its main function and components which far more complex and expensive than are finally needed. In the same way that the production model presents a cheaper and simpler version of the prototype, refactored code presents a simpler version of the initial code, which is easier to comprehend and thus cheaper to maintain.

2.4 Existing Conference Management Systems

There are currently a number of other conference management tools available to conference conveners. While they are predominately open-source, some are not. The use of PHP in development however is a definite trend in all newer projects however, along with the use of MySQL. The open source nature (and thus capital cost-free nature) of these software packages may go some way to explaining their popularity.

In addition to two of the systems listed below, another system similar to COMMENCE was presented to an international conference (Bogunović, Pek, Lončarić & Mornar 2003). The designers argued that PHP was more secure than the use of CGI scripts, which were used in prior conference management tools and had security issues.

However, with so many competitors with similar architectures, COMMENCE will need to distinguish itself from other systems to attract developers.

2.4.1 START (Version 2)

START could be seen as a market leader in conference management systems, and is the inspiration behind COMMENCE. As proprietary software, a working demo is unavailable without prior arrangement with its owner, *SoftConf.com*, nor are licensing or hosting prices. However, installation and feature documentation are available.

The structure of START can essentially be considered to be a more mature version of COMMENCE, with additional collaborative features for interaction between reviewers, and support for committee member functions. Before this thesis project, most differences were in areas of detail or additional sophistication rather than differences in goals.

One major difference between START and COMMENCE however is that while COMMENCE uses an external database which can be located on a different server to the web server, START does not require or allow use of an external database (Gerber 2003). This makes the hosting requirements for START more cumbersome, as files must be stored on the web server which the conference is hosted on. Such difficulties were one contributing factor to the development of COMMENCE.

2.4.2 SciCon^{PHP}

SciCon^{PHP} is a similar project to COMMENCE, and aims to be a conference management system for scientific conferences. It is maintained by Denice Deatrich of the Swiss Federal Institute of Technology, who works in the Information Theory and Communications Theory departments. As may be expected for two projects with similar goals from similar development environments, SciCon^{PHP} contains many common features to COMMENCE.

One major difference between SciCon^{PHP} and COMMENCE is the implementation and installation design philosophies. SciCon does not attempt to be an entire system that can be easily installed by an experienced user without programming experience. Instead, it has focused improving and adding functionality to become “feature-complete” rather than allowing users to “simply install it and use it without getting your hands dirty” (Deatrich 2003). The software is aimed at administration by a relatively experienced user of Unix-like systems who has skills

Figure 2.2: Screenshot of SciCon^{PHP}

with shell scripts, PHP and SQL. The stated opinions of the developer also suggest that easy installation and administration are not planned in the near future.

At the beginning of this thesis project, SciCon^{PHP} appeared to serve as a good example of the functionality that COMMENCE should eventually attain. Specifically, conference registration was an important feature which COMMENCE lacked. However, SciCon^{PHP} did not serve as a good example of how future functionality should be implemented in COMMENCE. While possibly more secure, its choice to use Apache for password protection severely limited deployment flexibility. Some database changes must be done by hand, and some mass email notification features were intentionally left unimplemented in favour of shell scripts. The extensive documentation, while comprehensive, was necessary to walk the administrator through the largely manual installation process, with heavy reliance on scripting to bring the installation time down to a reasonable length.

Overall, SciCon^{PHP} provided some concepts which may be useful in COMMENCE, although it did not serve as a good example for implementation methods. Additionally, SciCon^{PHP} development appears to have been inactive for over a year, indicating that it is unlikely to be a competitor with COMMENCE in the future.

2.4.3 WCMT

WCMT is another open source conference management system which bears some similarity to COMMENCE, which is being developed by Dr Andrea Rizzoli of the Dalle Molle Institute for Artificial Intelligence. The usability of WCMT is closer to COMMENCE than SciCon^{PHP}, however its simplicity of design, in both its use of Apache for user authentication and its lack of customization features, belies software which has been designed to solve the specific problem of the developer, rather than the general problem of conference management.

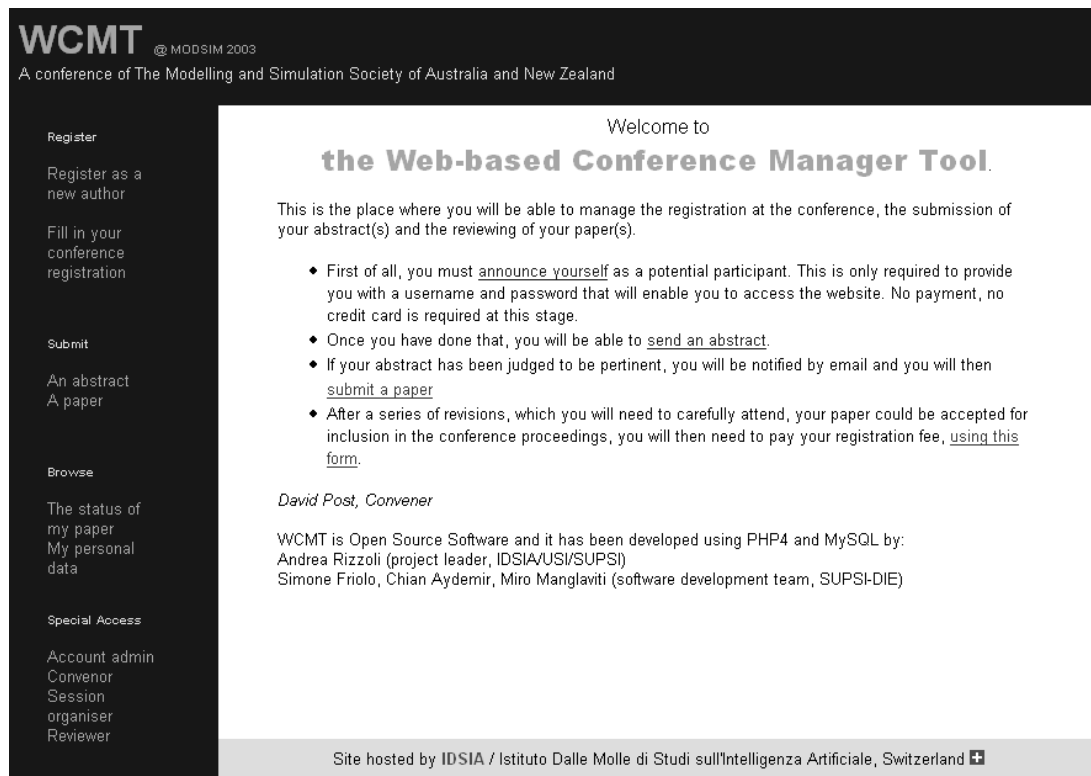



Figure 2.3: Screenshot of WCMT

WCMT is also written in PHP, and uses MySQL for storage, however unlike COMMENCE it uses frames for providing its menus. COMMENCE uses JavaScript menus to allow for environments where its pages may be contained in a frame themselves, and also because they provide more space due to their hierarchical nature.

2.4.4 OpenConf

Arguably the most similar conference management tool to COMMENCE is OpenConf. Developed by Zakon Group, OpenConf is a conference management system which provides “electronic submission, review and advocacy of papers, along with management of the whole process by the Program Chair” (Zakon Group LLC 2004a). Like other conference management systems, OpenConf is written in PHP and uses MySQL to store its data (Zakon Group LLC 2004a).

OpenConf Demo Conference 2005

OpenConf Conference Management System	
OpenConf Home	Email Chair 

Authors:

- Submit a Paper
- Edit Submission
- Re-Upload Paper
- Upload Final Paper

Review & Program Committees:

- Sign In
- Sign up

keycode required:

Program Chair:

- Sign In



Powered by OpenConf 
 Copyright ©2001-2004 Zakon Group LLC 

Figure 2.4: Screenshot of OpenConf

OpenConf proudly boasts that it has been used by over 50 conferences and journals (Zakon Group LLC 2004a) based around the world and including a number of Australian conferences (Zakon Group LLC 2004b). As an open source product aimed at appealing to a broad audience, OpenConf can also be seen as the main competitor for COMMENCE, as they both aim for a non-programmer market interested in open source conference management solutions.

Chapter 3

Planning for Expansion

3.1 Expanding, Extending and Improving COMMENCE

For COMMENCE to remain a viable open source project, it must continue to have useful features added, and attempt to distinguish itself from other projects like it. The main advantages of COMMENCE over other conference management systems before this thesis project were:

- the ability to use a database back-end on a separate machine to the web server;
- standalone authentication rather than web server authentication (eg. Apache);
- an administrator interface requiring no programming knowledge, and little technical expertise; and
- a flexible open source license.

However, COMMENCE did not stand out from other conference management systems in terms of features available, and some areas of its design were lacking. To conference conveners, COMMENCE was a more serious option than SciCon^{PHP} or WCMT, however both START V2 and OpenConf offered more polished systems (both claim to be mature systems, while COMMENCE does not) without any appreciable loss of functionality.

To rectify this, and to make COMMENCE better prepared for future development, discussions were held with the project founder Brian Lovell to ascertain how COMMENCE could be improved. From these discussions, six goals for the thesis project were decided upon, in order of priority:

1. Addition of post submission management, covering session scheduling, and automatic generation of conference programs and conference CD proceedings
2. Addition of conference registration for users
3. Better customization features, covering terminology and user registration forms
4. Migration of database calls to ADOdb
5. Better documentation

6. Refactoring of existing code

The goal for better documentation was put forward more tentatively than the others, as Brian Lovell felt that documentation for a project that was not yet stable would result in greater problems than benefits.

In the initial thesis project plan, the requirements for these goals were stated as follows:

- Post Submission Management
 - COMMENCE should contain a system for allocating accepted papers to sessions, and automatically allocating a slot inside that session to each paper using logical criteria.
 - The automatic session slot allocation system must allow VIP speakers to be allocated priority positions in the session, and should also allow for manual overriding if required.
 - An online and printable program of proceedings should be automatically generated by COMMENCE using session data.
 - A skeleton CD file structure should be produced automatically by COMMENCE, with page number references to the hard-copy paper compilation.
 - The paper compilation should be produced as a single PDF by COMMENCE, which can be used to produce the printed copy.
- Conference Registration
 - COMMENCE should allow the automatic generation of a registration form for printing and posting.
 - Registration options should be configurable by the convener.
 - Registration form should allow the user to choose from a variety of payment options. (Actual payment by these methods is not considered within the scope of COMMENCE.)
 - COMMENCE should allow the convener to confirm payment with COMMENCE after a transaction has taken place. Confirmation of payment and options should take place with the minimum required workload for the convener.
- Customization
 - The convener should be able to easily alter any terminology or that could reasonably be expected to vary between conferences.
 - The convener should be able to easily modify default and acceptable values for the registration form in cases where variation could be reasonably expected.

- ADOdb
 - All existing database interactions are to be rewritten to utilize the ADOdb library.
 - Any future database interactions are to utilize the ADOdb library.
 - The ADOdb library is to be integrated into the COMMENCE project either by the addition of the library itself, or by the addition of installation and integration instructions for the library.
- Documentation
 - All new functions should be accompanied by internal documentation of its purpose and a description of any input parameters and return values.
 - If an aforementioned function contains code from which the method cannot be easily discerned, then documentation of the method followed should be provided by means of either overall explanation or detailed comments.
 - All new files should contain appropriate descriptions of their function and any relevant licensing information.
 - New files shall not require change logs, however CVS commit comments should contain sufficient information to discern changes.
 - Basic installation and use instructions should be written and maintained. These instructions do not need to be comprehensive, but should serve as a quick start guide for new administrators.
- Refactoring
 - During the development of other features, refactoring of existing code structures may become necessary. Where refactoring does take place, notes will be made concerning the changes made, and a description of the refactoring which took place shall be included in the final report.

In addition to these goals, it was expected that the implementation of them would preserve the flexibility and ease of use already present in COMMENCE.

Chapter 4

Implementing the Plan

4.1 Improving the Overall Design

4.1.1 Restructuring the Settings Framework

One considerable problem with COMMENCE at the beginning of this thesis project was the lack of flexibility developers had when adding new features. COMMENCE had a facility for changing settings for the conference, such as the dimensions of the conference logo, maximum sizes for file uploads and the signature for conference emails. However, the ability to extend this facility was poor for a number of reasons.

Primarily, the problem was that not much thought had gone into designing the settings table. Initially, the settings table looked something like the diagram below.

Fields	MaxUploadSize	MaxLogoSize	EmailSignature	...
Row 1	10485760	1048576	'Thanks, Conference Management'	...

Figure 4.1: Original Settings Table

The result of this design was that to add new settings required a change to the structure of the database. From a development point of view, this meant that any addition to settings controlled by COMMENCE needed to be carefully thought out, because it would mean that any previous installations of COMMENCE would not be compatible with it without changing the table definitions too. The other problem this raised was that the functions which stored settings data were written with specific references to field names. So adding a new setting require changing a table definition, modifying the function which stored the settings data, and also adding a section to the settings page. As might be envisioned, this left developers disinclined to go to the effort.

To rectify this issue, the table format was changed to a basic (*name* \rightarrow *value*) structure. Instead of naming settings values through the fields, the settings were named through one of two fields in the new table, which was the primary key. In some ways, the new table format resembled a hash table data structure.

New settings could be added in this new table by simply inserting a new row, greatly increasing the ease at which new settings could be added. With the addition of code which dynamically decided on field names from the settings form itself, the amount of code required to add a new

Fields	Name	Value
Row 1	MaxUploadSize	10485760
Row 2	MaxLogoSize	1048576
Row 3	EmailSignature	'Thanks, Conference Management'
Row N

Figure 4.2: New Settings Table

setting was reduced to the entry on the settings form itself.

One problem with this design however was the need to use the same data type for each field, thus removing any special handling for dates and numbers. However, as PHP uses dynamic typing, where variables can shift between string and numeric values depending on the context in which they are used, this was unlikely to cause any major inconvenience to developers. The name of the settings were stored as a 128 character string, and the value field was set to use the data type `BLOB`. Ordinarily used for binary data, such as images, this data type preserved string formatting and allowed entries of any size easily conceivable for a conference setting. This had the additional benefit, although not originally foreseen, of allowing files to be uploaded from the settings page into the settings table, which was useful in the implementation of registration features.

4.1.2 Refactoring the User Registration Form

COMMENCE previously included user registration forms for the administrator, reviewers and normal users, for use in relation to the submitting and reviewing of papers. Unfortunately, there were a number of problems with the way this was implemented. Each registration form was nearly identical, however each one existed as part of the page it was displayed on, so any change to the form format required making changes to all of those files. Additionally, the function which processed the form fields and inserted them into the database used each field as an argument, meaning that any oversight in changing all the files would result in errors for those files not changed.

The solution to this was to amalgamate all of the registration forms into one function which could be called to display a registration form. Additionally, the database insertion function was modified so that it would process an dictionary of keys and values to obtain the field values to be added. This resulted in any changes to the registration form only requiring one change to the registration form display function, rather than multiple changes. For changes affecting the data stored however (such as adding a field), the database definition would still need to be changed.

4.1.3 Standardizing the Date Format

One particularly bad design decision that had been made in COMMENCE was the choice of internal date format. The format that was used was of the European format “Day-Month-Year”, and all dates extracted from the database were required to be in that format. Problems arose when Americanized dates were required (ie. “Month-Day-Year”) as the format needed to be changed in every case that a date was used. Date validation was a particularly bad problem, as dates valid under an American data format would not be valid under a European format. These

problems were further complicated by requirement for multiple date formats to suit difference conferences.

The solution implemented was to convert the internal date system to the calendar date portion of the ISO 8601 time format - “Year-Month-Day” (International Organization for Standardization 2003). All dates were to be input in this format, as dates such as “2004-07-04” could not be confused for either the American or European systems. In addition, a date converter function was written to turn dates in this format into any that could be represented by PHP.

Finally, a JavaScript pop-up calendar was used for fields requiring date input, both for ease of use and to remove any remaining confusion. The calendar was sourced from a free code website, with appropriate consideration taken to ensuring that the code license was compatible with that of COMMENCE.

The screenshot shows a web form with the following fields: Start Date, End Date, Location, Host Name, Contact Email, and Logo for Conference. The Start Date field contains '2004-05-06' and the End Date field contains '2004-12-20'. Below these fields are 'Submit' and 'Undo Changes' buttons. A JavaScript calendar is displayed over the End Date field, showing the month of December 2004. The calendar has a grid with days of the week (Mon to Sun) and dates (1 to 31). The date 20 is highlighted. At the bottom of the calendar, it says 'Today is Sat, 23 Oct 2004'.

Figure 4.3: JavaScript calendar and date input fields

4.1.4 Improving Customization and Localization

Utilizing the changes made to the settings and registration forms, customization of COMMENCE was substantially improved. Previously, terms which could change between conferences were stored as global variables in the COMMENCE source code, and were changed as conferences required. However, with an easily customizable settings page and a universal user registration form, changes to these settings had the potential to be made much easier. To cite examples of this, changes to the name for a paper’s primary topic, or to the default country for the registration form, are now performed through the settings page, and the global variable associated with them has been substituted for a reference to those particular settings.

4.1.5 Database access with ADOdb

Currently, the only database atop which COMMENCE can be run is MySQL. However, MySQL is not the only database server used by institutions wishing to host conferences. Support for databases such as Oracle would be preferable for the future success of the COMMENCE project.

The ADOdb library provides a standard interface for interaction with many different database servers. Support currently includes “MySQL, Oracle, Microsoft SQL Server, Sybase, Sybase SQL Anywhere, Informix, PostgreSQL, FrontBase, SQLite, Interbase (Firebird and Borland variants), Foxpro, Access, ADO, DB2, SAP DB and ODBC” (Lim 2004). Using the

Home	Conference ▼	Papers ▼	User Admin ▼	Program ▼	Registration ▼	General Admin ▼	Logout
------	--------------	----------	--------------	-----------	----------------	-----------------	--------

Settings

General Settings

Website Homepage:

Upload Settings

Max Upload Paper File Size MB
For uploaded papers

Maximum Logo File Size MB
For uploaded logo file

Maximum Logo Dimensions x
(Height x Width in pixels)
Logos larger than these dimensions will be resized.

Email Settings

Email Signature: (optional)
This text will be included at the end of all outgoing email.

Thanks, Conference Management

Localization Settings

Primary Topic Term
A paper can have only one primary topic.

Secondary Topics Term
A paper can have many secondary topics in addition to its primary topic.

Level Term

Default Country

Short Time Format

Long Time Format

Figure 4.4: New COMMENCE Settings page

ADODB library, COMMENCE could remove its reliance on a particular database server and make itself accessible to a wider range of users and developers.

Unfortunately, multiple database product support is not as simple to implement as simply using ADODB. Each different database server uses different data types to store information, and creating tables with generic data types is not possible using ADODB at this stage. The scope for this thesis project was to implement ADODB in COMMENCE, so that it would be easier to add support for other databases in future.

Firstly, the ADODB libraries were added to the COMMENCE code base. Calls were then changed over to the new format for both the wrapper function used to establish connections, and all functions making database calls. SQL queries also had to be changed to allow the ADODB libraries to correctly quote strings, and to use the correct date formats when inserting and selecting data. Methods checking and storing user passwords also had to be changed to use the SHA1 hashing function in PHP, rather than the MySQL specific `PASSWORD()` function. While labour intensive, this work was straightforward in most cases.

One area of PHP that was brought to light while changing over all the database calls was the “Magic-Quotes” feature. “Magic-Quotes” automatically processes strings in PHP and adds escaping slashes so that they can be used in a quoted expression without causing problems. The most common form of “Magic-Quotes” is `magic_quotes_gpc` which uses magic quotes for **G**et, **P**ost and **C**ookie data. While “Magic-Quotes” may sound useful, one problem with it is that it can be enabled or disabled by the server administrator, but not at runtime. Thus while some

PHP installations use “Magic-Quotes”, others do not. In developing COMMENCE this means that a test must be made for “Magic-Quotes” before adding escaping slashes to a string. The `db_quote` wrapper function implemented in changing to ADOdb does just that, and is used to escape (or leave if already escaped) strings for use in SQL queries.

4.1.6 Migrating to UTF-8

At the beginning of this thesis project, all pages were displayed using the ISO 8859-1 character set, which is Latin alphabet text. While this may have been previously acceptable, region-based character coding does not allow for some symbols which some other regions take for granted. For instance, the euro symbol (€) does not exist in ISO 8859-1 consistently. With the introduction of XML in COMMENCE (as discussed in the later section on conference registration), converting COMMENCE to use the recommended *text/xml* character encoding of UTF-8 made sense.

UTF-8 is a compromise format between UTF-16 (Unicode) and previous forms of character encoding. Unlike Unicode, UTF-8 only uses only one byte for standard ASCII text instead of two, which makes it backwards compatible with previous character sets. UTF-8 represents the rest of the Unicode characters using multi-byte characters of up to 4 bytes in length. Conversion was simply performed by altering the COMMENCE page template to output UTF-8 as the content encoding in outgoing headers. There was no need to convert individual documents to UTF-8, as they only used standard ASCII text, and thus were perfectly acceptable as UTF-8 pages.

4.1.7 Improving HTML code

Previously, COMMENCE employed tables in a large number of pages for layout purposes. This is now considered an improper way to design page layouts, with the introduction and increasing acceptance of HTML designed for the specific purpose of implementing layouts, such as division (`<div>`) and spanning (``) tags, and Cascading Style Sheets (CSS). In areas where it was feasible, these new layout methods were used when adding or altering pages in COMMENCE. In addition, the page template for COMMENCE was revised to use these newer layout methods, thus reducing the complexity of the code and improving the page aesthetics.

4.2 Post-Submission Management

4.2.1 Overview

The aim of Post-Submission Management (PSM) was to create a scheduling system for COMMENCE, and then produce features which drew upon that scheduling data. Below is an overview of the entities in COMMENCE PSM, and the relationships between them.

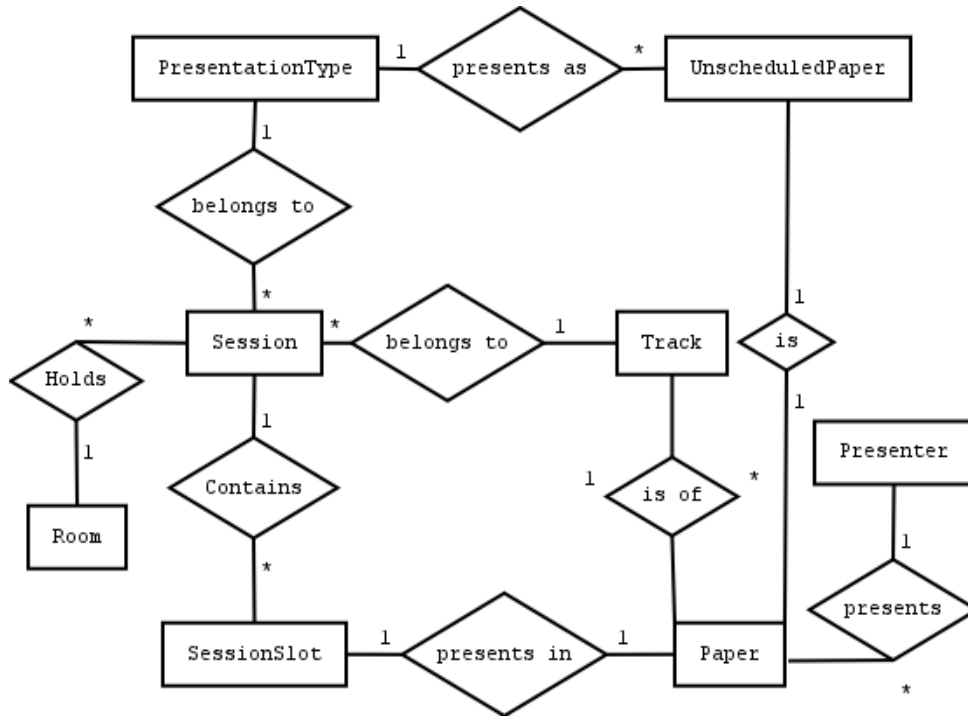


Figure 4.5: Relationships between entities in PSM

When a paper is accepted, it is assigned a presentation type (eg. Oral, Poster, etc.), and appears as an entry in the `UnscheduledPaper` table. As a paper, it also has a track (or primary topic). Papers are only allowed to be presented in Sessions which match their track and presentation type. Each presentation type has a set amount of time required to present it. Combined with the start and finish times of a session, this provides the number of papers that can be presented in a session, and thus the number of session slots. When a paper is scheduled, it is removed from the `UnscheduledPaper` table and given a session slot in a compatible session. That session slot has an ID denoting which number slot it is in its session. In addition, sessions must be held in rooms, and papers must have a designated presenter.

See Appendix A for the full list of table creation definitions.

4.2.2 Rooms and Presentation Types

The first and simplest step was to create pages to add rooms and presentation types for the conference. For simplicity and flexibility, rooms only had a name and an automatically assigned ID number. For presentation types the case was similar, with the addition of the time required for presenting a paper assigned to the presentation type.

4.2.3 Sessions

When first deciding what information to store as part of a session, there was a fundamental question that needed to be resolved: “Did the length of the session determine the number of session slots, or did the number of session slots determine the length?”. If it were the former, then the start time and finish time of the session would need to be set, but the number of slots would be determined by the session’s presentation type. If it were the latter, then the number of session slots would need to be set in addition to the start time, and this would determine the finish time.

The decision on this question was eventually decided in relation to flexibility. Specifying the start and finish time of a session allowed the convener to add some time padding to the session if they felt that some slots would not run to time, where as the allocation of a number of sessions would not allow for this. As both solutions would work, the decision was made to give the convener this extra flexibility.

A similar decision was made in relation to the start and finish dates. While conventional wisdom would dictate that a session should not finish on a day other than it started, there was no reason to restrict the convener in this regard. This decision should probably be reviewed after PSM has been used in actual conferences, as the convener may be prepared to forgo such flexibility in favour of not needing to select two fields values.

Sessions each belong to a certain track (otherwise known as a primary topic), which determines the papers that may be presented in them. While the possibility for not allocating a track was available (so that any paper could be allocated to the session) this option was not implemented as there was insufficient reason to believe it would be needed, and such a decision would have entailed a much more complex automatic scheduling algorithm.

After a new session is added, the auto-scheduler is invoked to allocate any unscheduled papers of the correct type to the new session.

Home	Conference ▾	Papers ▾	User Admin ▾	Program ▾	Registration ▾	General Admin ▾	Logout
------	--------------	----------	--------------	-----------	----------------	-----------------	--------

Add Session

Session Name	<input type="text" value="Demo Session"/>
Track	<input type="text" value="Pattern Recognition"/> ▾
Presentation Type	<input type="text" value="Poster"/> ▾
Start Time	<input type="text" value="2004-10-21"/> @ <input type="text" value="9"/> : <input type="text" value="00"/> <input type="text" value="am"/>
Finish Time	<input type="text" value="2004-10-21"/> @ <input type="text" value="12"/> : <input type="text" value="00"/> <input type="text" value="pm"/>
Room	<input type="text" value="1st Demo Room"/> ▾
Chairperson's Username	<input type="text" value="Ponder"/> ▾
<input type="button" value="Submit"/>	

About Commence

Figure 4.6: The Add Session page

4.2.4 Modifying the Accept/Reject Paper page and Letter

Before the implementation of PSM, the conference convener could decide after viewing the reviewer ratings for a paper whether to accept or reject the paper. This page was changed to include the selection of a presentation type as well. In doing this, it became obvious that

the acceptance letter would also need to be changed. The COMMENCE code for both of these features showed serious signs of requiring refactoring should further changes be required, however due to time constraints these deficiencies were left for a future developer to rectify, as they would essentially require rewriting all the existing letter-writing code.

Upon accepting a paper as a certain presentation type, the paper is added to the `UnscheduledPaper` table. At this point, the auto-scheduler is invoked to attempt scheduling the paper.

4.2.5 Automatically Scheduling Sessions

When the auto-scheduler function is invoked, it checks all the papers in `UnscheduledPaper` to see if it can allocate them to a session. It performs a query of all the sessions which match the paper's track and presentation type, and orders them by start time. It then progressively checks for sessions which do not have their full number of session slots allocated. Where it finds space for a new slot, it allocates the paper to it, and removes the paper from the `UnscheduledPaper` table.

4.2.6 Rescheduling Slots Manually

The problem of rescheduling session slots was difficult as it required user input on items in a database, yet also a high level of responsiveness to the convener using the system.

Previously in COMMENCE, all processing of pages relating to databases was done on the server side. If a change was selected in a form, the server would need to be notified before the page would change. A good example of this is the paper submission interface, where the number of authors selected in the form changes the number of fields for author information, but to do so the server must send a new form page with the extra fields in it. While this behaviour is appropriate in that situation, the action of refreshing the page from the server every time a session slot is shifted to another position could get quite tedious.

Home	Conference ▾	Papers ▾	User Admin ▾	Program ▾	Registration ▾	General Admin ▾	Logout
------	--------------	----------	--------------	-----------	----------------	-----------------	--------

Rearrange Session Slots

Track: Pattern Recognition
Presentation Type: Poster

#7 Identifying the hard lessons from...
 #6 The Second Journey to the Undersid...
 Free Slot
 Free Slot
 Free Slot
 Free Slot

Free Slot
 Free Slot
 Free Slot

▲
▼

Change

- ◆ Presenter has paid towards registration
- ◇ Presenter has not paid any registration
- Presenter has not been assigned

Ponder's Session
2nd Demo Room
Fri, 23 Jul 2004 @ 9:00 am to 1:00 pm

Ponder's Second Session
1st Demo Room
Sat, 24 Jul 2004 @ 3:00 pm to 5:00 pm

About Commence

Figure 4.7: Manually rearranging session slots in COMMENCE

Instead, the session slot reallocation page uses JavaScript to manipulate the positions of

session slots in a series of list boxes, and then sends the final rearranged slot positions to the server for processing. This means the convener can manipulate session slot positions in real-time, and then commit their choices all at once. This also has the benefit of allowing the convener to experiment with different orderings without changing the original schedule. The code which performs this appears in Appendix B.

The session slot reallocation page allows slots to be moved between sessions of the same type, in addition to moving them inside their own session. Colouring is used to indicate the presenter's registration status, so the convener can allocate low risk paper presentations (where the presenter has already paid) earlier slots in the sessions and higher risk paper presentations (where the presenter has not registered, or has not been assigned) to slots later in the sessions. This functionality was suggested by Brian Lovell, as he noted that presenters that had already paid a registration fee had a far higher probability of attending the conference, and that it would be preferable for missing presentations to occur towards the end of a session rather than the beginning.

4.2.7 Conference Program

Using the input session data, a conference program is generated by COMMENCE for use by the convener, with distribution of the data at the convener's discretion.

An example is shown in Appendix C.

4.2.8 Generating the Conference CD structure

The conference CD structure contains two web pages and all of the papers accepted and scheduled in the conference. The first web page contains a technical program, in which all of the papers are sorted in their order of presentation under day and session headings. The second web page is a list of all the authors who have written papers for the conference, and their papers listed by the page number they would have in the printed copy of the proceedings. The page number is extrapolated by taking the papers in presentation order, and using their page length to calculate which number each would start on, ensuring all start on odd numbered pages as they would in the printed copy.

Home	Conference ▼	Papers ▼	User Admin ▼	Program ▼	Registration ▼	General Admin ▼	Logout
------	--------------	----------	--------------	-----------	----------------	-----------------	--------

Build CD Structure

Select the type of archive should be exported:

☒ bzip2 (*.tar.bz2)

☐ gzip (*.tar.gz)

☐ zip (*.zip)

Base filename:

(conference name will be used if none given)

Build Archive

Figure 4.8: Building the CD Structure in COMMENCE

The CD structure may be downloaded in BZip2, GZip or Zip formats. The three different formats are given because each has its own strength:

- **BZip2** offered the best compression by a significant margin during testing, and is widely available for many platforms;
- **GZip** can be extracted on practically any Unix-like system; and
- **Zip** can be extracted natively under Windows XP.

Once the convener has selected their choice in archive format, COMMENCE begins building the archive. Because this can be a long process, a progress bar is displayed.

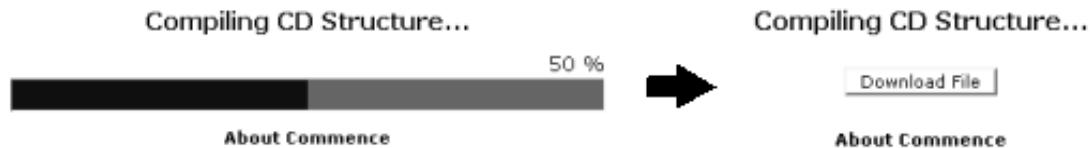


Figure 4.9: Compiling and Downloading the CD Structure in COMMENCE

The libraries for displaying a progress bar were sourced from the PHP Extension and Application Repository (PEAR), and their PHP license is compatible with their inclusion in the COMMENCE source code.

4.3 Attendee Registration

4.3.1 Overview

The goal of implementing a conference registration form for COMMENCE was to enable the conference convener to track the statistical details of registrations in a simple and effective manner. Previously, manual counts needed to be kept on the details of conference registrations, and a total figure of moneys paid generally required a request to the convening organisation's finance department. The online registration form for COMMENCE removes both of those hassles, and allows the convener to track the attendance of presenters for Post-Submission Management.

COMMENCE already implemented a system for user registration, which is used for submitting papers for review and admission to the conference. As this account type is available to the general public, conference registration functionality was added to it with the view that it be used by all persons wishing to attend the conference, rather than those simply submitting papers.

One of the key difficulties with designing the registration system was the need to be able to completely customize the registration form. This prevented storing the conference registration data directly into the database, as database definitions cannot be easily changed. Even a more flexible table structure, such as the one used with the settings table, would have been fraught with problems, as effectively querying such a table for detailed data would be troublesome at best.

What was needed was a second layer of information storage which could be easily customized but also queried. Fortunately, such a technology already existed: XML.

4.3.2 XML and XSLT

XML (Extensible Markup Language) describes XML documents, which are text documents composed of data storage units called entities (Bray, Paoli, Sperberg-McQueen, Maler & Yergeau 2004). These entities can be used to store data in way which is both human- and machine-readable. XSLT (Extensible Stylesheet Language Transformations) is a language which can be used to transform XML documents into other XML documents (Clark 1999), including XHTML documents. In combination, XML and XSLT provide a way to store data and transform it into viewable web pages.

The PHP interpreter includes as an optional component an XSLT engine. This allows XML documents to be queried and transformed using PHP and an XSLT stylesheet. Using an XML definition to define the conference registration form, an XSLT stylesheet is used to transform that definition into a online form in COMMENCE. By storing that data received from that form as XML as well, it can be combined by other XSLT stylesheets to produce registration statements and statistics.

4.3.3 The Registration Form XML Definition

To logically portray a conference registration form, a definition of the format needed to be devised. The formal DTD (Document Type Definition) and more verbose "XML Registration Form Specification" for form designers are attached as Appendices D & E respectively, with an

example definition included in Appendix F. For convenience however, the fundamentals of the registration form definition will be outlined here.

A registration form is composed of four fundamental entities: payment times, registration types, memberships and choices.

Payment times are different time periods in which a conference attendee can pay their registration. Most conferences give an “early bird” discount to attendees who pay early, which serves as a way of increasing cash flow earlier in conference organisation. The attendee must pick the payment time covering when they expect to pay their registration, which will determine the initial prices of all items in the registration form.

COMMENCE does not concern itself with any form of date checking for payment times, as to do so would remove convener discretion from such matters without adding much in the way of functionality. It is assumed that the convener will check the user has selected the correct payment time when processing their registration fee.

Payment Time

- ☒ **Early Bird** - Before 1-Jun-98
- ☐ **Later** - On or after 1-Jun-98

Figure 4.10: Payment times in a COMMENCE conference registration form

Registration types represent a type of registration as would normally be displayed on a traditional conference registration form, such as “Full” or “Student”. Each registration type has a price for each payment time, so that options such as “Early Full Registration” consist of selecting the “Full” registration type and the “Early” payment time.

Each registration type can have included items, such as a copy of the conference proceedings or tickets to the conference dinner. These items are composed of “choice” selections, which occur later in the registration form.

Registration Type

- ☒ **Full Conference Registration** - Includes lunches, coffee breaks.

Includes:

- 1x Conference Dinner
- 1x Welcoming Cocktail Party
- 1x Conference Proceedings

Price		Discounts	
<i>Early Bird</i>	AUD\$ 730.00	<i>IAPR Member</i>	AUD\$ 36.50
<i>Later</i>	AUD\$ 880.00	<i>IEEE Member</i>	AUD\$ 14.60

- ☐ **Student Conference Registration** - Includes lunches, coffee breaks.

Includes:

- 1x Conference Proceedings

Price		Discounts	
<i>Early Bird</i>	AUD\$ 580.00	<i>IAPR Member</i>	AUD\$ 29.00
<i>Later</i>	AUD\$ 730.00	<i>IEEE Member</i>	AUD\$ 11.60

Figure 4.11: Registration types in a COMMENCE conference registration form

Memberships are professional memberships that entitle conference attendees to discounts on items in the registration form, including the initial registration fee. For instance, a conference convenor may choose to give a discount of \$40 on the full registration fee to all IEEE members wishing to attend the conference.

Membership

☐ **IAPR Member** - Member of the International Association for Pattern Recognition

☒ **IEEE Member** - Member of the Institute of Electrical and Electronics Engineers

Figure 4.12: Memberships in a COMMENCE conference registration form

Choices are extra items (which may also be included with some registration types) that can be purchased by the attendee in addition to their initial registration fee. These items can range from dinner tickets to conference tutorials to T-shirts. There are two types of choice: those with options, and those with a quantity.

Choices with options consist of a number of different selections which the user can make, of which they can select a maximum of one. These options have a price attached to them for each payment time and may also have discounts linked to memberships or the selected registration type. When the price is the same for all payment times, and there are no discounts, only the price is shown to the prospective attendee.

Morning Session

☒ **none**

☐ **Mathematical Morphology for Image Analysis and Pattern Recognition** - Presenter is Pierre Soille.

Price	Discounts
<i>Early Bird</i> AUD\$ 150.00	<i>Student Conference Registration</i> AUD\$ 50.00
<i>Later</i> AUD\$ 175.00	

☐ **Robust Techniques for Computer Vision** - Presenter is Peter Meer.

Price	Discounts
<i>Early Bird</i> AUD\$ 150.00	<i>Student Conference Registration</i> AUD\$ 50.00
<i>Later</i> AUD\$ 175.00	

Figure 4.13: A choice with options in a COMMENCE conference registration form

Choices with a quantity only represent one type of selection, but one which may be selected a number of times. For instance, the number of tickets for a conference dinner would fit into this type of choice. Such choices can have a minimum and maximum allowed number of items, have an price for each payment time and may also have discounts linked to memberships or the selected registration type. The displayed pricing information is however limited to the bare minimum that user needs to see, with multiple prices or discounts only displayed if they exist.

Welcoming Cocktail Party

When paying registration, please include a list of guests and company for the printing of extra name tags.

0	at the cost per each of	Price
		AUD\$ 25.00

Figure 4.14: A choice with a quantity in a COMMENCE conference registration form

4.3.4 Conference Registration Statement

Using the selections made by the user in the conference registration form, in combination with the registration form definition, a statement is generated by another XSLT stylesheet. It contains an itemized list of all the selections made, with appropriate pricing and discounting of items. The attendee is then asked to confirm that this selection is correct, upon which time they are issued instructions on how to use that form and its unique identification number to make payment to the convening organisation.

An attendee may fill out as many different forms as they like, and they need only forward the one they wish to pay to the conference convener. Unpaid and paid forms alike are permanently stored in the COMMENCE database as a section of XML representing the selections made by the user and the total price for the form. This XML is attached to the registration form XML and transformed by an XSLT stylesheet whenever a conference statement needs to be displayed.

An example conference registration statement is shown in Appendix G.

4.3.5 Verifying Conference Registration Forms

Conference form verification by the convener was surprisingly simple to implement, and is very simple to use. The administrator only needs to type in the unique ID number of the paid registration form to display a copy of the statement and an indicator as to whether the form has been paid or not. The payment status can be changed quickly and simply.

Home	Conference ▼	Papers ▼	User Admin ▼	Program ▼	Registration ▼	General Admin ▼	Logout
------	--------------	----------	--------------	-----------	----------------	-----------------	--------

Form #4

This form is currently marked as **unpaid** . Change this to :

☐ Paid
☒ Unpaid

Michael William Bailes (*mitchrawnurodo*)
University of Queensland
Ph: 33788827
Email: bailes@uq.net.au

Payment Time : **Early Bird**
Before 1-Jun-98

Relevant Memberships :

- ♦ IAPR Member

Choice	Selection	Price
Registration Type	Full Conference Registration Includes lunches, coffee breaks	AUD\$ 693.50

Figure 4.15: Verifying the payment of a registration form in COMMENCE

In running a conference, cases do occur where the registration form needs to be changed, either because payment was not done inside the selected payment time, or because the attendee's

requirements have changed. In those cases, the easiest option for the conference convener is to request the user fill out a new form, to mark any previously paid form as unpaid again, and to mark as paid the new form once the attendee has submitted it.

4.3.6 Registration Statistics

Just as conference registration statements are generated using the XML selection data and the XML form definition, so too are the conference statistics. COMMENCE extracts all the selection XML for paid statements, and then an XSLT stylesheet tallies that data against the XML registration form definition to generate a breakdown of all paid conference registrations, and the total moneys those registrations must have resulted in being paid.

An example of the statistics page is shown in Appendix H.

4.3.7 Changing Registration Form Details

The XML registration form definition contains most of the data required for generating a registration form, however it deliberately does not include some information. The currency symbol used in the registration form is one such case, as the convener may wish to change the way it is displayed without changing the registration form. Likewise, the preamble to the registration form, and the instructions for what should be done afterwards, are instead controlled through COMMENCE's settings page.

Replacing the XML registration form definition can be done from the settings page through a file upload field. This is in keeping with COMMENCE's emphasis on keeping all settings in the database, rather than in the file-system.

Registration Settings

<p>Show Conference Registration Features <i>Conference registration features can be hidden from end-users by selecting "No" here. These features should only be enabled after an XML registration form template has been uploaded.</i></p>	<input checked="" type="checkbox"/> Yes <input type="checkbox"/> No
<p>Registration Preamble <i>This text will be displayed at the top of the conference registration form. It should include any information that the registrant is likely to need that is not on the form itself.</i></p>	<div style="border: 1px solid black; padding: 5px; min-height: 50px;"> All prices are in Australian Dollars (AUD). </div>
<p>Currency Unit <i>This is the currency unit that will displayed next to all prices on the form.</i></p>	<div style="display: flex; align-items: center;"> <input type="text" value="AUD\$"/> <div style="margin-left: 10px;"> <input checked="" type="checkbox"/> Before <input type="checkbox"/> After </div> </div>
<p>Registration Form XML <i>The XML definition for the registration form.</i></p>	<div style="display: flex; align-items: center;"> <input type="text"/> <input type="button" value="Browse..."/> </div>
<p>Registration Final Instructions <i>This text will be displayed after the user has completed a conference registration form, telling them what to do to complete the registration.</i></p>	<div style="border: 1px solid black; padding: 5px; min-height: 100px;"> <pre><p> To complete your conference registration, send a copy of the registration statement, payment and any relevant membership details to: </p> <ul type="none"> Address 1 ... </pre> </div>

Figure 4.16: Changing the registration settings in COMMENCE

Chapter 5

Results of Work

The initial goals of the thesis project, as outlined in the project plan, and their outcomes are addressed below in relation to their success or failure at the end of this thesis project. In some cases, the intention of the goal was successfully achieved, although not in the way initially envisaged. Rationale for considering those cases a success is included, along with an analysis of failures.

5.1 Post-Submission Management

- ✓ COMMENCE should contain a system for allocating accepted papers to sessions, and automatically allocating a slot inside that session to each paper using logical criteria.

COMMENCE now contains functionality which automatically allocates papers to the first available session and session slot. The “logical criteria” referred to in the goal were initially envisaged to include the payment status of the presenter, however it was decided through consultation with Brian Lovell that allocation to the first available slot would be preferable, with sorting on basis of payment status carried out manually.

- ✓ The automatic session slot allocation system must allow VIP speakers to be allocated priority positions in the session, and should also allow for manual overriding if required.

Manual reallocation of sessions is available through the session slot reallocation page. Instead of distinguishing between VIP and ordinary presenters, it was decided leave such decisions to manual session slot allocation instead. One reason for this is that in the case of multiple sessions, different conferences may have different ways of allocating VIP speakers to sessions. While different options for VIP speaker allocation could have been made available, a far simpler and more time effective approach in this case was to rely on manual session slot allocation to meet this objective.

- ✓ An online and printable program of proceedings should be automatically generated by COMMENCE using session data.

An online conference program is now available to the convener. A decision was made against making the conference program available to users, as the dissemination of such information before the timetable is finalised could lead to confusion. In retrospect, this could have been handled in the same way as registration form access - a simple switch in the conference settings used to turn access to the conference program on and off. Implementation of this would be fairly straightforward.

A printable program for the conference was not produced, as the online program was believed to be of a high enough standard not to warrant the requirement of a second “for-print” version. While such a feature could be implemented in future, it was decided that this was not of a high enough priority to divert time from other goals.

- ✓ A skeleton CD file structure should be produced automatically by COMMENCE, with page number references to the hard-copy paper compilation.

This goal was carried out successfully, and an additional technical program was included in the CD structure.

- ✗ The compilation should be produced as a single PDF by COMMENCE, which can be used to produce the printed copy.

This goal was not completed, although for reasons which were deemed acceptable. The decision to include this goal was made before performing adequate research into how it could be achieved. On conducting this research at a latter stage, it was discovered that there currently exists no platform independent, freely available and consistent way of merging PDF documents. All of these attributes are vital to maintaining the vision of COMMENCE development, and without this capability available, creating a single PDF document of all conference papers was not possible. Implementing such a system was dismissed on the grounds of the time available for completion of this thesis project, and that external processes already exist to compile such a document using Adobe software packages.

5.2 Conference Registration

- ✓ COMMENCE should allow the automatic generation of a registration form for printing and posting. Registration options should be configurable by the convener.

This goal was completed in entirety by using XML to supplement the COMMENCE database structure. Initially, it had been envisioned that manipulation of any form of code would not be required, however the very flexibility provided in the registration system prevented the timely implementation of such an administrator interface. While this does to a certain extent lower the ease of use for COMMENCE, it can be argued that the necessary skills in editing XML would not be beyond the average conference convener with the aid of detailed documentation. Such documentation was developed, although time to evaluate its effectiveness was not available.

- ✓ Registration form should allow the user to choose from a variety of payment options. (Actual payment by these methods is not considered within the scope of COMMENCE.)

As the the conference registration system is currently implemented in COMMENCE, there is no specific functionality that allows for the specification of payment options. Such functionality could however be provided through the “choices” available in a registration form, or less formally in the customizable guides to completing conference registration.

- ✓ COMMENCE should allow the convener to confirm payment with COMMENCE after a transaction has taken place. Confirmation of payment and options should take place with the minimum required workload for the convener.

COMMENCE now includes payment confirmation features for conference registration, with forms accessible quickly and easily via their unique identification number.

5.3 Customization

- ✓ The convener should be able to easily alter any terminology or that could reasonably be expected to vary between conferences.

The COMMENCE settings page allows the terms for tracks, topics and levels to be altered, as well as redefinition of the names for primary and secondary email and phone details. This resolves all currently known areas of concern for terminology customization. Values that are deemed in future to require customization can easily be added to the settings page through the improved settings framework.

- ? The convener should be able to easily modify default and acceptable values for the registration form in cases where variation could be reasonably expected.

The initial area of concern in the registration form, in relation to the default country selection, has now been resolved. However, these customization alterations were performed relatively early in the thesis project, and other settings such as phone number and postcode formats may fit into the definition of “variation reasonably expected” for acceptable values. Under these circumstances, while this goal has not failed in its initially conceived form, it has not been completely met either.

5.4 ADOdb

- ✓ All existing database interactions are to be rewritten to utilize the ADOdb library.

All database interactions now utilize the ADOdb library.

- ✓ Any future database interactions are to utilize the ADOdb library.

All database interactions implemented in the course of this thesis project utilize the ADOdb library.

- ✓ The ADOdb library is to be integrated into the COMMENCE project either by the addition of the library itself, or by the addition of installation and integration instructions for the library.

The ADOdb library was directly included into the COMMENCE code base, which simplifies installation.

5.5 Documentation

- ✗ All new functions should be accompanied by internal documentation of its purpose and a description of any input parameters and return values.

This internal documentation standard was not met. To some extent, this occurred because such documentation would be required for every function to be consistent. However, for the most part this was simply a case of neglecting documentation due to time constraints.

- ? If an aforementioned function contains code from which the method cannot be easily discerned, then documentation of the method followed should be provided by means of either overall explanation or detailed comments.

Unfortunately, the definition of “easily discerned” is not well defined, and code which is straightforward to some may not be to others. Attempts were made to explain sections of code which were considered tricky by the author. However, for a majority of cases, newly implemented code relies on being simple enough to understand without commenting, rather than having complete comments on its function.

- ✗ All new files should contain appropriate descriptions of their function and any relevant licensing information.

One concern at the beginning of this project was that from a strict interpretation of the MPL, all COMMENCE files needed to have MPL headers attached to them to be covered by the license. However, like other documentation goals, this was not performed, largely because such a change was not considered urgent by project leader Brian Lovell.

- ✓ New files shall not require change logs, however CVS commit comments should contain sufficient information to discern changes.

In all commits to the COMMENCE CVS server, a good attempt was made to provide concise descriptions of the changes made to the COMMECNE code.

- ✗ Basic installation and use instructions should be written and maintained. These instructions do not need to be comprehensive, but should serve as a quick start guide for new administrators.

Due to time constraints, the installation process and its documentation were not addressed in this thesis project.

5.6 Refactoring

- ✓ During the development of other features, refactoring of existing code structures may become necessary. Where refactoring does take place, notes will be made concerning the changes made, and a description of the refactoring which took place shall be included in the final report.

Substantial refactoring was performed as part of the thesis project, with the descriptions included in this document.

Chapter 6

Future Development

This thesis project highlighted a number of areas for future work and development of COMMENCE. Four of these areas deal with rectifying existing issues with COMMENCE's design, while the fifth relates to changes which could significantly improve COMMENCE compared to other conference management tools.

6.1 PHP5 Compatibility

On 13 July 2004, PHP 5.0.0 was released (PHP.net 2004). Since then, two bug-fix releases have followed, with the rectification of a number of serious flaws (PHP.net 2004). However, PHP5 will eventually become commonplace, and COMMENCE will need to ensure that it is compatible with PHP5. Currently, COMMENCE uses a deprecated feature of PHP called “call-time pass-by-reference” for a considerable number of its functions. While not currently an insurmountable problem for running COMMENCE with PHP5, it does require certain warning notices to be disabled in the PHP5 configuration files before COMMENCE can be run. This, and other issues that may come to light, need to be resolved in the near future or COMMENCE may suffer in terms of its user base and developer interest.

6.2 Letter Composition

Currently, COMMENCE allows the generation of a number of letters for user notification. However, the framework used to generate these letters is very inflexible and in need of replacement for COMMENCE to be comparable with other conference management systems. All letters are currently the product of custom PHP code, and adding additional letters requires serious effort on the part of the developer. Ideally, a system would be implemented that allowed the conference convener to create their own letters using a wide variety of data from COMMENCE, including data from the settings table.

6.3 Database Transactions

A transaction is a logical unit of database processing, such as the insertion of information about a technical paper, that includes one or more database access operations (Elmasri & Navathe 2000).

In the case of a paper, this may mean the insertion of records into several tables, possibly using existing values in those tables.

Currently, COMMENCE does not implement transactions for any of its database queries, insertions or updates. For systems which have few concurrent users, this does not present too many problems. However, for large conferences this can lead to serious problems, as queries may take place against data which is in the process of being changed. For COMMENCE to be ready for large conferences, it will need to implement transactions for almost all of its queries.

ADODB provides functionality for using transactions, and MySQL can use tables that implement transactions too. Unfortunately, ensuring that MySQL is totally safe from concurrent database queries and modifications would require a significant amount of effort.

6.4 Increased registration form flexibility

While the COMMENCE registration form functionality is currently fairly advanced, there are still some additional features conferences may want in their registration forms. Percentage discounting is one such area, as currently all discounts exist as a flat rate. In addition, some conferences like to allow “special case” prices which do not meet any formula of discounts, but are simply a flat fee for people meeting a set of criteria.

The current system of payment times determining the initial prices may also be an area which needs improvement. The decision to use this solution was made before it became apparent that discounts based on registration type would also be needed. A more generalized solution which may work better would be to have an initial price irrespective of any criteria, and instead to incorporate payment times as criteria for discounts.

A wider review of existing traditional conference registration forms may prove helpful in these matters.

6.5 Using XML more broadly in COMMENCE

Currently many sections of COMMENCE are more rigid than they could be for conveners. User registration forms (as opposed to conference registration forms) have a set format of fields, which may not suit all conferences. Likewise, paper submissions also place restrictions on the sort of data that can be obtained.

The experience with implementing a conference registration form has shown that XML can be used to add significant flexibility in conference management systems. Extending the use of XML to other areas of COMMENCE would empower conference conveners to modify COMMENCE to suit the specific needs of their conference, while removing some of the burden from developers to allow for all possible options.

COMMENCE forms could store data in a combination of database fields and XML entities. Data used to identify rows in SQL queries would be stored in database fields, allowing the searching of records to be done efficiently by the relational database server. Data stored only for informational purposes - such as paper abstracts, reviewer comments and user contact details - could be stored as XML in a field with those records. When the form is displayed, the field

data and XML would be seamlessly combined, allowing the convener to track any informational data they might want, while still allowing fast record searching via the relational database.

However, such changes to COMMENCE would also need a standard system for altering XML form definitions inside COMMENCE, as imposing increasing burdens on conveners to learn XML would run counter the established aims of the COMMENCE project. The current absence of such a system for conference registration form design is due to time constraints on development, rather than a fundamental shift in ideals, and it is expected that COMMENCE will eventually have such a system implemented. Solving the more general case of altering XML form definitions in COMMENCE rather than the specific case of the conference registration form would allow COMMENCE to obtain significantly higher flexibility.

Chapter 7

Conclusion

With the completion of the work outlined in this document, COMMENCE is now capable of managing a conference from the point of the request for paper submissions, all the way to the first day of the conference. No other conference management system discovered during background research contains these capabilities.

With the addition of Post Submission Management, conference conveners can now plan their conferences programs as papers are approved. With the addition of conference registration through COMMENCE, conveners can track registrations and their moneys received, allowing them to make estimates for purchasing and printing decisions. COMMENCE still remains relatively easy to use, despite having greatly increased in its scope and functionality.

Refactoring of the COMMENCE source code has yielded greater flexibility for developers at precisely the same time that COMMENCE has increased its scope and added new features. This should have a positive impact on its appeal to open source developers. While XML and XSLT skills are now required for development of registration features, this has potentially increased the available skill base, as development of registration features can be performed without any database or PHP skills. For developers with a background in XML technologies, this may provide a new avenue for contributions.

There is still much work to be done, and further challenges ahead, in preparing COMMENCE for Brian Lovell's goal of large scale conference management by 2008. PHP5 compatibility is a high priority to prevent the increasing user base of that software becoming disenchanted with COMMENCE. However, great inroads have been made towards this goal, and hopefully future developers will succeed in making COMMENCE the leader in its field.

Bibliography

- Behlendorf, B. (1999), Open source as a business strategy, in ‘Voices from the Open Source Revolution’, O’Reilly & Associates. <http://www.sindominio.net/biblioweb/telematica/open-sources-html/main.html>.
- Bogunović, H., Pek, E., Lončarić, S. & Mornar, V. (2003), An electronic journal management system, in ‘25th International Conference on Information Technology Interfaces’, Vol. 16, pp. 231–236.
- Bray, T., Paoli, J., Sperberg-McQueen, C. M., Maler, E. & Yergeau, F. (2004), *Extensible Markup Language (XML) 1.0*, third edn, W3C. <http://www.w3.org/TR/REC-xml/>.
- Clark, J. (1999), *XSL Transformations (XSLT) Version 1.0*, W3C. <http://www.w3.org/TR/xslt>.
- Crowston, K. & Scozzi, B. (2002), Open source software projects as virtual organisations: competency rallying for software development, in ‘IEEE Proceedings - Software’, Vol. 149.
- Deatrich, D. C. (2003), *SciCon^{PHP} Installation Document*. <http://scicon-php.sourceforge.net/doc/sciconphp-install/node2.html>.
- Elmasri, R. & Navathe, S. B. (2000), *Fundamentals of Database Systems*, third edn, Addison-Wesley, p. 631.
- Fowler, M. (1999), *Refactoring : improving the design of existing code*, Addison-Wesley.
- Gerber, R. (2003), *Start V2 Installation Documentation*. <http://www.softconf.com/install.html>.
- International Organization for Standardization (2003), ‘Numeric representation of dates and time’. <http://www.iso.ch/iso/en/prods-services/popstds/datesandtime.html>.
- Lim, J. (2004), *ADODB Library for PHP*. <http://phplens.com/lens/adodb/docs-adodb.htm>.
- Open Source Initiative (2004), ‘Open Source Initiative OSI - Welcome’. <http://www.opensource.org/>.
- PHP.net (2004), ‘Php 5 changelog’. <http://www.php.net/ChangeLog-5.php>.
- Zakon Group LLC (2004a), ‘OpenConf - The Friendly Conference Management System’. <http://www.zakongroup.com/technology/openconf.shtml>.

Zakon Group LLC (2004*b*), 'OpenConf Portfolio'. <http://www.zakongroup.com/technology/openconf-conferences.shtml>.

Appendix A

PSM Database Definitions

CREATE TABLE Session

```
(
    /* Session ID of a session being held */
    SessionID SMALLINT AUTO_INCREMENT PRIMARY KEY ,
    /* Session name */
    SessionName TINYTEXT NOT NULL ,
    /* The track all papers in this session belong to, NULL if no restriction */
    TrackID TINYINT UNSIGNED,
    /* Presentation type ID of that one session */
    PresentationTypeID TINYINT UNSIGNED NOT NULL,
    /* Start time of session */
    StartTime DATETIME NOT NULL ,
    /* End time of session */
    EndTime DATETIME NOT NULL ,
    /* Room ID where conference is held */
    RoomID TINYINT UNSIGNED NOT NULL ,
    /* RegisterID of Reviewer who is chairing the conference at one session */
    ChairID INT UNSIGNED NOT NULL
);
```

CREATE TABLE SessionSlot

```
(
    /* Session ID of a session the slot is in */
    SessionID SMALLINT NOT NULL,
    /* Slot ID of which dictates order in slot */
    SlotID SMALLINT UNSIGNED NOT NULL,
    /* Paper ID of presented paper */
    PaperID INT UNSIGNED NOT NULL,
    PRIMARY KEY ( PaperID )
);
```

CREATE TABLE Room

```
(
    /* Room ID of a place where session being held */
    RoomID TINYINT UNSIGNED AUTO_INCREMENT PRIMARY KEY ,
    /* Room where conference is held
    eg. Chapel 01 / Common Room 02 / Chapel Gardens 03 */
    RoomName TINYTEXT NOT NULL
);
```

```
CREATE TABLE Presenter
```

```
(  
    /* Paper ID which the presenter would talk about */  
    PaperID INT UNSIGNED NOT NULL PRIMARY KEY,  
    /* Registered ID of a presenter at a slot */  
    RegisterID INT UNSIGNED NOT NULL  
)  
;
```

```
CREATE TABLE PresentationType
```

```
(  
    /* Presentation Type ID */  
    PresentationTypeID TINYINT AUTO_INCREMENT PRIMARY KEY ,  
    /* eg, Oral, Poster, etc. */  
    PresentationTypeName VARCHAR(30) NOT NULL,  
    /* Length in minutes of a SessionSlot of this type */  
    SlotLength INT NOT NULL  
)  
;
```

```
/* Holds papers accepted but not in sessions yet */
```

```
CREATE TABLE UnscheduledPaper
```

```
(  
    PaperID INT UNSIGNED NOT NULL PRIMARY KEY,  
    PresentationTypeID TINYINT NOT NULL  
)  
;
```


Appendix B

Session Rescheduling JavaScript

```
function show_properties(obj)
{
    for (element in obj){
        document.write(element+"<br />");
    }
}

function movedown(listbox)
{
    var lb1 = -1; // index of selected listbox
    for (i = 0; i < listbox.length; i++)
    {
        if (listbox[i].selectedIndex > -1)
        {
            lb1 = i;
        }
    }
    // this is the list box of the to-be-swapped cell
    var lb2 = lb1;
    // if index is too high, go to next box
    if ((listbox[lb1].selectedIndex + 1) == listbox[lb1].length)
        lb2 = (lb1 + 1) % listbox.length;

    var lbs1 = listbox[lb1].selectedIndex;
    var lbs2 = (lbs1 + 1) % listbox[lb1].length ;

    // Swap text
    var text = listbox[lb1].options[lbs1].text;
    listbox[lb1].options[lbs1].text = listbox[lb2].options[lbs2].text;
    listbox[lb2].options[lbs2].text = text;

    // Swap value
    var value = listbox[lb1].options[lbs1].value;
    listbox[lb1].options[lbs1].value = listbox[lb2].options[lbs2].value;
    listbox[lb2].options[lbs2].value = value;

    // Swap colour
    var color = listbox[lb1].options[lbs1].style['color'];
    listbox[lb1].options[lbs1].style['color'] = listbox[lb2].options[lbs2].style[
        'color'];
    listbox[lb2].options[lbs2].style['color'] = color;

    listbox[lb1].selectedIndex = -1;
```

```

        listbox[lb2].selectedIndex = lbs2;
    }

    function moveup(listbox)
    {
        var lb1 = -1; // index of selected listbox
        for (i = 0; i < listbox.length; i++)
        {
            if (listbox[i].selectedIndex > -1)
            {
                lb1 = i;
            }
        }
        // this is the list box of the to-be-swapped cell
        var lb2 = lb1;
        // if index is too low, go to previous box
        if (listbox[lb1].selectedIndex - 1 < 0)
            lb2 = (lb1 + listbox.length - 1) % listbox.length;

        var lbs1 = listbox[lb1].selectedIndex;
        var lbs2 = lbs1 - 1;
        if (lbs2 < 0)
            lbs2 = listbox[lb2].length - 1;

        // Swap text
        var text = listbox[lb1].options[lbs1].text;
        listbox[lb1].options[lbs1].text = listbox[lb2].options[lbs2].text;
        listbox[lb2].options[lbs2].text = text;

        // Swap value
        var value = listbox[lb1].options[lbs1].value;
        listbox[lb1].options[lbs1].value = listbox[lb2].options[lbs2].value;
        listbox[lb2].options[lbs2].value = value;

        // Swap colour
        var color = listbox[lb1].options[lbs1].style['color'];
        listbox[lb1].options[lbs1].style['color'] = listbox[lb2].options[lbs2].style[
            'color'];
        listbox[lb2].options[lbs2].style['color'] = color;

        listbox[lb1].selectedIndex = -1;
        listbox[lb2].selectedIndex = lbs2;
    }

    function clearothers(theList, allLists)
    {
        for (var i = 0; i < allLists.length; i++)
        {
            if (allLists[i].id != theList.id)
            {
                allLists[i].selectedIndex = -1;
            }
        }
    }

    function process(f)
    {

```

```
var result = "";
for (var i = 0; i < f.ListBox.length; i++)
{
    if (f.ListBox[i].nodeName != "SELECT") continue;
    result += f.ListBox[i].id+"= ";
    for (var j = 0; j < f.ListBox[i].length; j++)
    {
        result += f.ListBox[i].options[j].value+" ";
    }
    result += " ";
}
f.NewOrder.value = result;
}
```


Appendix C

Example Conference Program

Home	Conference ▾	Papers ▾	User Admin ▾	Program ▾	Registration ▾	General Admin ▾	Logout
------	--------------	----------	--------------	-----------	----------------	-----------------	--------

Conference Program

Friday, 23 July 2004

Pattern Recognition

Computer Vision

Medical Imaging

Applications

9:00 am - 1:00 pm
2nd Demo Room
Ponder's Session
#7 Identifying the hard leassons from projects - easily
#6 The Second Journey to the Underside of the Disc

Saturday, 24 July 2004

Pattern Recognition

Computer Vision

Medical Imaging

Applications

3:00 pm - 5:00 pm
1st Demo Room
Ponder's Second Session

Friday, 30 July 2004

Pattern Recognition

Computer Vision

Medical Imaging

Applications

9:00 am - 11:00 am
3rd Demo Room
CompVis

12:00 pm - 1:00 pm
3rd Demo Room
CompVis2

9:00 am - 10:00 am
2nd Demo Room
MedImg

10:00 am - 12:00 pm
2nd Demo Room
Apps

Wednesday, 4 August 2004

Pattern Recognition

Computer Vision

Medical Imaging

Applications

9:00 am - 11:00 am
1st Demo Room
OralCompVis
#2 Journey to the Underside of the Disc

Thursday, 21 October 2004

Pattern Recognition

Computer Vision

Medical Imaging

Applications

9:00 am - 11:00 am
4th Demo Room
Demo Session

About Commence

Figure C.1: Generated conference program in COMMENCE

Appendix D

Registration Form DTD

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

<!DOCTYPE orderform [
  <!ELEMENT orderform (regtypes,paytimes,memberships?,section+) >

  <!-- ID types -->
  <!ELEMENT paytimeid (#PCDATA) >
  <!ELEMENT regtypeid (#PCDATA) >
  <!ELEMENT memberid (#PCDATA) >
  <!ELEMENT detailid (#PCDATA) >
  <!ELEMENT choiceid (#PCDATA) >
  <!ELEMENT optionid (#PCDATA) >
  <!ELEMENT quantityid (#PCDATA) >

  <!-- Basics -->
  <!ELEMENT name (#PCDATA) >
  <!ELEMENT description (#PCDATA) >
  <!ELEMENT cost (#PCDATA) >
  <!ELEMENT number (#PCDATA) >
  <!ELEMENT minimum (#PCDATA) >
  <!ELEMENT maximum (#PCDATA) >

  <!-- Registration Types -->
  <!ELEMENT regtypes (regtype+) >
  <!ELEMENT regtype (regtypeid,name,description,price*,discount*,included?) >
  <!ELEMENT included (includedchoice*) >
  <!ELEMENT includedchoice (choiceid, (optionid—number)) >

  <!-- Payment Times -->
  <!ELEMENT paytimes (paytime+) >
  <!ELEMENT paytime (paytimeid,name,description) >

  <!-- Professional Memberships -->
  <!ELEMENT memberships (membership+) >
  <!ELEMENT membership (memberid,name,description) >

  <!-- Section (used to give heirarchies) -->
  <!ELEMENT section (section*,choice*,detail*) >

  <!-- Detail (a text entry) -->
  <!ELEMENT detail (detailid, name, description?) >

  <!-- Choices -->
  <!ELEMENT choice (choiceid,name,description?,(option+—quantity)) >
  <!ELEMENT option (optionid,name,description?,price*,discount*,(minimum,
    maximum?)) >
  <!ELEMENT quantity (price*,discount*,(minimum,maximum?)) >
```

```
<!ATTLIST quantity minimum CDATA "0" >  
<!ATTLIST quantity maximum CDATA "1" >  
<!ELEMENT price (paytimeid,cost) >  
<!ELEMENT discount ((regtypeid—memberid),cost) >
```

]>

Appendix E

XML Registration Form Specification

Common Components

< regtypeid >, < paytimeid >, < memberid >, < choiceid >, < optionid >

Unique ID for the parent element in question. Any references to the parent are with the ID, so they should not be changed/removed once registration has started.

< name >

The displayed name of the parent element. This can be safely changed after registration has begun.

< description >

The displayed description of the parent element. This can be safely changed after registration has begun.

< number >

The a number of items, used in relation to included choices.

< cost >

Monetary value. Used by **<price>** to convey a value owed by the attendee to the conference, and used in **<discount>** to convey an amount paid by the conference to the attendee.

< price >

children: **<paytimeid>**, **<cost>**

Provides a cost for the given time of payment. For instance, *early* may have a cost of \$100, but *late* may have a cost of \$150. Therefore, the item would have two price elements, as below:

————— Price for *early* & *late* —————

```
<price>
  <paytimeid> early </paytimeid>
  <cost> 100.00 </cost>
</price>
<price>
  <paytimeid> late </paytimeid>
```

```

      <cost> 150.00 </cost>
</price>

```

< *discount* >

children: <memberid>, <regtypeid>, <cost>

Provides the amount to discount if the attendee holds the given membership or has the specified registration type. If an attendee is entitled to more than one discount for an item because they hold multiple memberships they will only receive the most beneficial one. If an attendee is eligible for a discount based on registration type, they still will receive any applicable membership discounts.

For instance, an item with an *early* cost of \$100, which is discounted \$20 if the attendee is a member of *myorg* would be as follows:

A discounted price

```

<price>
  <paytimeid> early </paytimeid>
  <cost> 100.00 </cost>
</price>
<discount>
  <memberid> myorg </memberid>
  <cost> 20.00 </cost>
</discount>

```

Registration Types and Modifiers

< *regtypes* >

children: <regtype>

Contains all the registration types for the form.

< *regtype* >

children: <regtypeid>, <name>, <description>, <price>, <discount>, <included>

A registration type - eg *Academic* or *Student*. It must contain one <price> for every <paytime>. It can contain multiple discounts. The <included> element is optional, and is used to convey items which normally would cost extra, but are included free in the cost of registration.

< *paytimes* >

children: <paytime>

Payment times for the form.

< *paytime* >

children: <paytime>, <name>, <description>

A payment time. In most conferences, an early-bird discount is given for items to encourage early registration. Likewise, certain items may cost more if registration is done late. Each item which has a cost associated with it must have a <price> for each <paytime>. Registration types are included in this, for example:

Two payment times and two registration types

```

<regtypes>
  <regtype>
    <regtypeid>academic</regtypeid>
    <name>Academic</name>
    <description>Is an academic</description>
    <price>
      <paytimeid>early</paytimeid>
      <cost>100.00</cost>
    </price>
    <price>
      <paytimeid>normal</paytimeid>
      <cost>130.00</cost>
    </price>
  </regtype>
  <regtype>
    <regtypeid>student</regtypeid>
    <name>Student</name>
    <description>Is a student</description>
    <price>
      <paytimeid>early</paytimeid>
      <cost>120.00</cost>
    </price>
    <price>
      <paytimeid>normal</paytimeid>
      <cost>150.00</cost>
    </price>
  </regtype>
</regtypes>
<paytimes>
  <paytime>
    <paytimeid>early</paytimeid>
    <name>Early</name>
    <description>Payment before YYYY-MM-DD</description>
  </paytime>
  <paytime>
    <paytimeid>normal</paytimeid>
    <name>Normal</name>
    <description>Normal payment time</description>
  </paytime>
</paytimes>

```

< memberships >

children: <membership>

Memberships affecting whether the attendee can get a discount.

< membership >

children: <memberid>, <name>, <description>

A memberships which may affect whether the attendee can get a discount. There is no side effect to having a membership without a single discount attached to it. When a discount is offered for an item, it refers to the <memberid> of the relevant membership.

section*s, Choices, Options and Quantities

< section >*

children: <name>, <description>, <choice>, <section*>

section*s can be used to break up choices that need to be made into logical parts. They have no effect on anything except the display of the form. section*s may be nested, and choices may exist at the same level as sessions.

< choice >

children: <choiceid>, <name>, <description>, <option>, <quantity>

A choice is a decision the attendee must make concerning either options or quantities. A choice may have options or a quantity, but not both. A choice with only one option will allow that option to be selected or not selected. A choice with multiple options will force one option to be selected. A choice with a quantity will allow a number to be selected between the quantity's maximum and minimum (inclusive).

Some examples are shown below, although keep in mind that the <option> and <quantity> tags are not complete.

_____ A choice between the red or the blue tie _____

```
<choice>
  <choiceid> ties </name>
  <name> Ties </name>
  <description> Which tie do you want to wear? </description>
  <option>
    <optionid> red </optionid>
    <name> The red one. </name>
  </option>
  <option>
    <optionid> blue </optionid>
    <name> The blue one. </name>
  </option>
</choice>
```

Sometimes you just want to know how much of something somebody wants, so you can use a quantity instead of options.

_____ Quantity of booklets _____

```
<choice>
  <choiceid>booklets</choiceid>
  <name>Booklets</name>
  <description> How many booklets would you like? </description>
  <quantity minimum="1" maximum="4">
    <price>
      <paytimeid>early</paytimeid>
      <cost>5.00</cost>
    </price>
    <price>
      <paytimeid>normal</paytimeid>
      <cost>8.00</cost>
    </price>
  </quantity>
</choice>
```

< option >

children: <optionid>, <name>, <description>, <price>, <discount>

An option is an item which has a price for each registration type, and possible discounts. It should also have a name and a description.

< quantity >

children: <price>, <discount>

attributes: minimum => Number, maximum => Number

A quantity represents a choice in the number of items, with prices and discounts attached on a per item basis. A minimum and maximum attribute should generally be set, although the default values are 0 and 1 respectively.

< includedchoice >

children: <choiceid>, <optionid>, <number>

An included choice is one which ordinarily would cost money but is included free as part of the registration fee. Depending on the type of choice, there may be one or multiple options listed, or a single number to denote the included quantity. For example:

Included items
<pre> <included> <includedchoice> <choiceid>tie</choiceid> <optionid>red</optionid> </includedchoice> <includedchoice> <choiceid>booklets</choiceid> <number>3</number> </includedchoice> </included> </pre>

Appendix F

Example Conference Registration Form Definition

```
<?xml version="1.0" encoding="UTF-8"?>
  <!--
  <!DOCTYPE orderform SYSTEM "../includes/regform.dtd">
  -->
  <orderform>
    <regtypes>
      <regtype>
        <regtypeid>full</regtypeid>
        <name>Full Conference Registration</name>
        <description>
          Includes lunches, coffee breaks.
        </description>
        <included>
          <includedchoice>
            <choiceid>dinner</choiceid>
            <number>1</number>
          </includedchoice>
          <includedchoice>
            <choiceid>cocktail</choiceid>
            <number>1</number>
          </includedchoice>
          <includedchoice>
            <choiceid>proceedings</choiceid>
            <number>1</number>
          </includedchoice>
        </included>
        <price>
          <paytimeid>early</paytimeid>
          <cost>730.00</cost>
        </price>
        <price>
          <paytimeid>late</paytimeid>
          <cost>880.00</cost>
        </price>
        <discount>
          <memberid>iapr</memberid>
          <cost>36.50</cost>
        </discount>
        <discount>
          <memberid>ieee</memberid>
          <cost>14.60</cost>
        </discount>
      </regtype>
      <regtype>
        <regtypeid>student</regtypeid>
```

```

<name>Student Conference Registration</name>
<description>
  Includes lunches, coffee breaks.
</description>
<included>
  <includedchoice>
    <choiceid>proceedings</choiceid>
    <number>1</number>
  </includedchoice>
</included>
<price>
  <paytimeid>early</paytimeid>
  <cost>580.00</cost>
</price>
<price>
  <paytimeid>late</paytimeid>
  <cost>730.00</cost>
</price>
<discount>
  <memberid>iapr</memberid>
  <cost>29.00</cost>
</discount>
<discount>
  <memberid>ieee</memberid>
  <cost>11.60</cost>
</discount>
</regtype>
<regtype>
  <regtypeid>daymon</regtypeid>
  <name>Day Registration – Monday</name>
  <description>
    Includes lunches and coffee breaks.
  </description>
  <price>
    <paytimeid>early</paytimeid>
    <cost>200.00</cost>
  </price>
  <price>
    <paytimeid>late</paytimeid>
    <cost>200.00</cost>
  </price>
  <discount>
    <memberid>iapr</memberid>
    <cost>10.00</cost>
  </discount>
  <discount>
    <memberid>ieee</memberid>
    <cost>4.00</cost>
  </discount>
</regtype>
<regtype>
  <regtypeid>daytue</regtypeid>
  <name>Day Registration – Tuesday</name>
  <description>
    Includes lunches and coffee breaks.
  </description>
  <price>
    <paytimeid>early</paytimeid>
    <cost>200.00</cost>
  </price>
  <price>
    <paytimeid>late</paytimeid>
    <cost>200.00</cost>

```



```

</price>
<discount>
  <memberid>iapr</memberid>
  <cost>10.00</cost>
</discount>
<discount>
  <memberid>ieee</memberid>
  <cost>4.00</cost>
</discount>
</regtype>
<regtype>
  <regtypeid>daywed</regtypeid>
  <name>Day Registration - Wednesday</name>
  <description>
    Includes lunches and coffee breaks.
  </description>
  <price>
    <paytimeid>early</paytimeid>
    <cost>200.00</cost>
  </price>
  <price>
    <paytimeid>late</paytimeid>
    <cost>200.00</cost>
  </price>
  <discount>
    <memberid>iapr</memberid>
    <cost>10.00</cost>
  </discount>
  <discount>
    <memberid>ieee</memberid>
    <cost>4.00</cost>
  </discount>
</regtype>
<regtype>
  <regtypeid>daythu</regtypeid>
  <name>Day Registration - Thursday</name>
  <description>
    Includes lunches and coffee breaks.
  </description>
  <price>
    <paytimeid>early</paytimeid>
    <cost>200.00</cost>
  </price>
  <price>
    <paytimeid>late</paytimeid>
    <cost>200.00</cost>
  </price>
  <discount>
    <memberid>iapr</memberid>
    <cost>10.00</cost>
  </discount>
  <discount>
    <memberid>ieee</memberid>
    <cost>4.00</cost>
  </discount>
</regtype>
</regtypes>

<paytimes>
  <paytime>
    <paytimeid>early</paytimeid>
    <name>Early Bird</name>
    <description>Before 1-Jun-98</description>

```

```

</paytime>
<paytime>
  <paytimeid>late</paytimeid>
  <name>Later</name>
  <description>On or after 1-Jun-98</description>
</paytime>
</paytimes>

<memberships>
  <membership>
    <memberid>iapr</memberid>
    <name>IAPR Member</name>
    <description>
      Member of the International Association for Pattern Recognition
    </description>
  </membership>
  <membership>
    <memberid>ieee</memberid>
    <name>IEEE Member</name>
    <description>
      Member of the Institute of Electrical and Electronics Engineers
    </description>
  </membership>
</memberships>

<section>
  <name>Extras</name>
  <description>
    Tickets and items in addition to those included in registration.
  </description>
  <choice>
    <choiceid>dinner</choiceid>
    <name>Conference Dinner</name>
    <description>
      When paying registration, please include a list of guests and
      company for the printing of extra name tags.
    </description>
    <quantity maximum="100">
      <price>
        <paytimeid>early</paytimeid>
        <cost>80.00</cost>
      </price>
      <price>
        <paytimeid>late</paytimeid>
        <cost>80.00</cost>
      </price>
    </quantity>
  </choice>
  <choice>
    <choiceid>cocktail</choiceid>
    <name>Welcoming Cocktail Party</name>
    <description>
      When paying registration, please include a list of guests and
      company for the printing of extra name tags.
    </description>
    <quantity maximum="100">
      <price>
        <paytimeid>early</paytimeid>
        <cost>25.00</cost>
      </price>
      <price>
        <paytimeid>late</paytimeid>
        <cost>25.00</cost>
      </price>
    </quantity>
  </choice>

```

```

    </price>
  </quantity>
</choice>
<choice>
  <choiceid>proceedings</choiceid>
  <name>Conference Proceedings</name>
  <description>(Additional)</description>
  <quantity maximum="100">
    <price>
      <paytimeid>early</paytimeid>
      <cost>150.00</cost>
    </price>
    <price>
      <paytimeid>late</paytimeid>
      <cost>150.00</cost>
    </price>
  </quantity>
</choice>
<choice>
  <choiceid>proceedings pages</choiceid>
  <name>Additional Proceedings Pages</name>
  <description>
    Three pages have been allocated within the conference proceedings
    for your accepted paper. Additional pages may be inserted if
    required.
  </description>
  <quantity maximum="100">
    <price>
      <paytimeid>early</paytimeid>
      <cost>150.00</cost>
    </price>
    <price>
      <paytimeid>late</paytimeid>
      <cost>150.00</cost>
    </price>
  </quantity>
</choice>
</section>
<section>
  <name>Tutorials</name>
  <description>
    These tutorials are optional. Should you wish to attend any of the
    listed tutorials, please indicate below.
  </description>
  <choice>
    <choiceid>morningtutorials</choiceid>
    <name>Morning Session</name>
    <option>
      <optionid>none</optionid>
      <name>none</name>
    </option>
    <option>
      <optionid>1</optionid>
      <name>
        Mathematical Morphology for Image Analysis
        and Pattern Recognition
      </name>
      <description>
        Presenter is Pierre Soille.
      </description>
      <price>
        <paytimeid>early</paytimeid>
        <cost>150.00</cost>
      </price>
    </option>
  </choice>

```

```

    </price>
    <price>
      <paytimeid>late</paytimeid>
      <cost>175.00</cost>
    </price>
    <discount>
      <regtypeid>student</regtypeid>
      <cost>50.00</cost>
    </discount>
  </option>
  <option>
    <optionid>2</optionid>
    <name>
      Robust Techniques for Computer Vision
    </name>
    <description>
      Presenter is Peter Meer.
    </description>
    <price>
      <paytimeid>early</paytimeid>
      <cost>150.00</cost>
    </price>
    <price>
      <paytimeid>late</paytimeid>
      <cost>175.00</cost>
    </price>
    <discount>
      <regtypeid>student</regtypeid>
      <cost>50.00</cost>
    </discount>
  </option>
  <option>
    <optionid>3</optionid>
    <name>
      Geographic Information Systems
    </name>
    <description>
      Presenter is Hanan Samet.
    </description>
    <price>
      <paytimeid>early</paytimeid>
      <cost>150.00</cost>
    </price>
    <price>
      <paytimeid>late</paytimeid>
      <cost>175.00</cost>
    </price>
    <discount>
      <regtypeid>student</regtypeid>
      <cost>50.00</cost>
    </discount>
  </option>
</choice>
<choice>
  <choiceid>afternoontutorials</choiceid>
  <name>Afternoon Session</name>
  <option>
    <optionid>none</optionid>
    <name>none</name>
  </option>
  <option>
    <optionid>4</optionid>
    <name>

```

```

Minimum encoding length inference methods
</name>
<description>
Presenter is David Dowe.
</description>
<price>
  <paytimeid>early</paytimeid>
  <cost>150.00</cost>
</price>
<price>
  <paytimeid>late</paytimeid>
  <cost>175.00</cost>
</price>
<discount>
  <regtypeid>student</regtypeid>
  <cost>50.00</cost>
</discount>
</option>
<option>
  <optionid>5</optionid>
  <name>
Automated Biometrics
</name>
  <description>
Presenters are
Nalini Ratha,
Andrew Senior,
Ruud M. Bolle and
Anil K. Jain.
</description>
  <price>
    <paytimeid>early</paytimeid>
    <cost>150.00</cost>
  </price>
  <price>
    <paytimeid>late</paytimeid>
    <cost>175.00</cost>
  </price>
  <discount>
    <regtypeid>student</regtypeid>
    <cost>50.00</cost>
  </discount>
</option>
<option>
  <optionid>6</optionid>
  <name>
Augmented Reality:
Confluence of Computer Vision and Computer Graphics
</name>
  <description>
Presenter is Mihran Tuceryan.
</description>
  <price>
    <paytimeid>early</paytimeid>
    <cost>150.00</cost>
  </price>
  <price>
    <paytimeid>late</paytimeid>
    <cost>175.00</cost>
  </price>
  <discount>
    <regtypeid>student</regtypeid>
    <cost>50.00</cost>
  </discount>
</option>

```

```

    </discount>
  </option>
</choice>
<choice>
  <choiceid>fulldaytutorials</choiceid>
  <name>Full Day</name>
  <option>
    <optionid>7</optionid>
    <name>
      3D Objects – Representation, Recognition and Coding
    </name>
    <description>
      Presenters are Chitra Dorai and Patrick Flynn.
    </description>
    <price>
      <paytimeid>early</paytimeid>
      <cost>300.00</cost>
    </price>
    <price>
      <paytimeid>late</paytimeid>
      <cost>350.00</cost>
    </price>
    <discount>
      <regtypeid>student</regtypeid>
      <cost>100.00</cost>
    </discount>
  </option>
</choice>
</section>
</orderform>

```

Appendix G

Example Conference Registration Statement

Demo Conference Registration

#4

Michael William Bailes (*mitchrawnurodo*)
University of Queensland
Ph: 33788827
Email: bailes@uq.net.au

Payment Time :

Early Bird
Before 1-Jun-98

Relevant Memberships :

- ♦ IAPR Member

Choice	Selection	Price
Registration Type	Full Conference Registration Includes lunches, coffee breaks. Included: <ul style="list-style-type: none">♦ 1x Conference Dinner♦ 1x Welcoming Cocktail Party♦ 1x Conference Proceedings	AUD\$ 693.50
Conference Dinner When paying registration, please include a list of guests and company for the printing of extra name tags.	2 @ 80.00	AUD\$ 160.00
Morning Session	Mathematical Morphology for Image Analysis and Pattern Recognition Presenter is Pierre Soille.	AUD\$ 150.00
Afternoon Session	Augmented Reality: Confluence of Computer Vision and Computer Graphics Presenter is Mihran Tuceryan.	AUD\$ 150.00
Full Day	3D Objects - Representation, Recognition and Coding Presenters are Chitra Dorai and Patrick Flynn.	AUD\$ 300.00
Total		AUD\$ 1453.50

Figure G.1: Conference registration statement in COMMENCE

Appendix H

Example Conference Registration Statistics

Home	Conference	Papers	User Admin	Program	Registration	General Admin	Logout
Payment Statistics							
Registration Types							
Full Conference Registration						4	
Deluxe Conference Registration						0	
Student Conference Registration						1	
Day Registration - Monday						0	
Day Registration - Tuesday						0	
Day Registration - Wednesday						0	
Day Registration - Thursday						0	
Payment Times							
Early Bird						3	
Later						2	
Memberships							
IAPR Member						3	
IEEE Member						1	
Choices							
Conference Dinner						8	
Welcoming Cocktail Party						6	
Conference Proceedings						5	
Additional Proceedings Pages						0	
Morning Session	none					2	
	Mathematical Morphology for Image Analysis and Pattern Recognition					2	
	Robust Techniques for Computer Vision					1	
	Geographic Information Systems					0	
	none					2	
Afternoon Session	Minimum encoding length inference methods					0	
	Automated Biometrics					1	
	Augmented Reality: Confluence of Computer Vision and Computer Graphics					2	
Full Day	3D Objects - Representation, Recognition and Coding					2	
Total Amount Paid						AUD\$ 5610.50	
About Commence							

Figure H.1: Conference registration statistics in COMMENCE