

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

# Fostering inter institutional knowledge sharing among students

A thesis presented in partial fulfilment of the requirements for the degree of

Master of Philosophy  
in  
Computer Science

at Massey University, Palmerston North,  
New Zealand.

Øyvind Smestad

2006

## **Abstract**

The aim of this research was to develop a system to allow students from various tertiary educational institutions, taking similar courses, to share common discussion forums. This was seen as important to enable larger communities of learners which could help each other, encourage discussion and give exposure to a more diverse range of opinions. In general to create a system that would make it easier to find knowledge in large networks of information and people.

For this purpose a special discussion forum system was developed that can easily be integrated with institutions existing software systems for course management. The system handles the inter-institutional communications transparently, and was developed to be flexible in how it can be installed on various server configurations. The special features of the system allow students to specify the type of message they are contributing, and the system then uses this information to adapt the user interface. For instance, when a question is added, the system searches for possible answers in the existing knowledge base and displays them.

An evaluation of the system in three tertiary educational institutions in New Zealand showed positive feedback from students, indicating they would use a system like this if it was made available to them in their future studies. An evaluation among teachers also showed a generally positive response. In the evaluation of the system's automatic answer finding capabilities, it was identified that this functionality should be improved to increase the effectiveness of how the system identified and highlighted possible answers.

### **Acknowledgements**

I would like to thank my supervisors Professor Kinshuk and Associate Professor Elizabeth Kemp for continuous support throughout this project and for allowing me the opportunity to do this. I would also like to thank those involved with the Advanced Learning Technologies Research Centre (ALTRC), especially everyone involved with the eQuake project.

A special thanks goes to Nicola, Chris, Ilkka, Kathryn and Josiana for reviewing drafts of this thesis.

And thanks to friends and family for encouragement throughout the process!

Part of this research has been supported by the eLearning Collaborative Development Fund of the Tertiary Education Commission of New Zealand.



## Table of contents

ACKNOWLEDGEMENTS	III
TABLE OF CONTENTS	IV
LIST OF FIGURES	VII
LIST OF TABLES	VIII

<b>1 INTRODUCTION</b>	<b>1</b>
1.1 EQuAKE PROJECT	2
1.2 DISCUSSION FORUMS: AN INTRODUCTION	3
1.3 DISCUSSION FORUMS IN EDUCATIONAL CONTEXT	5
1.4 LIMITATIONS OF DISCUSSION FORUMS	6
1.5 LIMITATIONS OF DISCUSSION FORUMS IN EDUCATIONAL CONTEXT	8
1.6 RESEARCH QUESTIONS	10
1.7 THESIS STRUCTURE	10
<b>2 LITERATURE REVIEW</b>	<b>11</b>
2.1 KNOWLEDGE FORUM (CSILE)	11
2.2 FUTURE LEARNING ENVIRONMENT (FLE)	13
2.3 STUDENT ASSISTANT AGENT IN FLE	14
2.4 DISCOURSE ENVIRONMENT FOR SHADOW NETWORKSPACE	14
2.5 FORUMPLUS	16
2.6 FLASH FORUMS AND FORUMREADER	17
2.7 DYNAMIC FREQUENTLY ASKED QUESTIONS ENVIRONMENT (DFAQ)	18
2.8 FINDING KNOWLEDGE RESOURCES	19
2.9 SOCIAL CREATIVITY AND SOCIAL CAPITAL	19
<b>3 REQUIREMENTS ANALYSIS</b>	<b>21</b>
3.1 REQUIREMENTS IDENTIFIED FOR NEW FORUM SYSTEM	21
3.2 MESSAGE TYPES / THINKING TYPES	22
3.3 SELECTION OF MESSAGE TYPES	23
3.4 HOW STUDENTS USE DISCUSSION FORUMS	24
3.5 INTEGRATION WITH LEARNING MANAGEMENT SYSTEMS	26
<b>4 DEVELOPMENT PLANNING</b>	<b>28</b>
4.1 TECHNOLOGIES USED IN THE SYSTEM	28
4.1.1 WEB SERVICES	28
4.1.2 AGENTS	28
4.1.3 OBJECT RELATIONAL MAPPING (ORM) TOOLS	29
4.2 SURVEY OF EXISTING OPEN SOURCE FORUM SYSTEMS	29
4.3 OVERVIEW OF PROPOSED SYSTEM	31
4.4 DEVELOPMENT APPROACH	32
4.5 DEVELOPMENT ENVIRONMENT	33
4.6 TASK ALLOCATION	35
4.7 TEAM COMMUNICATION TOOLS	35

<b>5</b>	<b><u>EQUAKE ARCHITECTURE</u></b>	<b>37</b>
<b>5.1</b>	<b>SYSTEM ARCHITECTURE</b>	<b>37</b>
5.1.1	LMS PLUG-INS	38
5.1.2	NOTIFICATION	40
5.1.3	FORUM DISPLAY	41
5.1.4	PROXY AGENT	41
5.1.5	TUTOR PROXY AGENT	41
5.1.6	STUDENT PROXY AGENT	41
5.1.7	TARGET SELECTION AGENT	42
5.1.8	QUERY MONITORING AGENT	42
5.1.9	DISPATCHER	43
5.1.10	DATA SERVICE	43
5.1.11	DATAMODEL	43
5.1.12	INSTANCE CONFIGURATION DATA	43
<b>5.2</b>	<b>DATA ARCHITECTURE</b>	<b>44</b>
5.2.1	DATA MODEL	45
5.2.2	DATA ACCESS LAYER	47
<b>5.3</b>	<b>DEPLOYMENT ARCHITECTURE</b>	<b>47</b>
5.3.1	FLEXIBILITY OF POSSIBLE DEPLOYMENT	50
<b>6</b>	<b><u>DISCUSSION OF IMPLEMENTATION ISSUES</u></b>	<b>52</b>
6.1	DISCURSIVE LEARNING ENVIRONMENT OR INTELLIGENT HELP SYSTEM	52
6.2	INTEGRATING MESSAGE TYPES IN THE USER INTERFACE	54
6.3	NAVIGATION OF LARGE MESSAGE ARCHIVES	54
<b>7</b>	<b><u>EVALUATION</u></b>	<b>56</b>
7.1	CONTINUOUS EVALUATION	56
7.2	HOW THE SYSTEM WAS SET UP FOR FORMAL USER EVALUATION	56
7.3	PARTICIPANTS	57
7.4	METHODOLOGY	58
7.4.1	TECHNOLOGY ACCEPTANCE MODEL	58
7.5	PROCEDURE	59
7.6	RESULTS	60
7.7	DISCUSSION OF USER EVALUATION	62
7.8	EVALUATION BY TEACHERS	64
<b>8</b>	<b><u>CONCLUSION</u></b>	<b>68</b>
8.1	IMPROVING ANSWER FINDING SEARCH RESULTS	70
8.2	FURTHER WORK	73
	<b><u>REFERENCES</u></b>	<b>77</b>
	<b><u>APPENDIX A: DATA DEFINITIONS</u></b>	<b>90</b>
<b>A.1</b>	<b>GLOBAL DATA SECTION:</b>	<b>90</b>

A.2 LOCAL DATA SECTION:	90
<b><u>APPENDIX B: EVALUATION FORMS</u></b>	<b>93</b>
B.1 EQUAKE EVALUATION QUESTIONNAIRE FOR USERS	93
B.2 EQUAKE EVALUATION QUESTIONNAIRE FOR STAFF	96
<b><u>APPENDIX C: EVALUATION RESULTS</u></b>	<b>98</b>
C.1 USER EVALUATION RESULTS	98
C.2 TEACHER EVALUATION RESULTS	99
<b><u>APPENDIX D: LIST OF TERMS</u></b>	<b>100</b>

## List of figures

FIGURE 1: THINKING TYPES FOR SNS DISCOURSE ENVIRONMENT	15
FIGURE 2: FORUMPLUS USER INTERFACE (FINI, 2006)	16
FIGURE 3: HIGH LEVEL SYSTEM OVERVIEW	31
FIGURE 4: USE OF TRADITIONAL FORUMS VS. EQUAKE FORUMS	37
FIGURE 5: SYSTEM DIAGRAM	38
FIGURE 6: CONTROL FLOW WHEN LOGGING IN TO AN EQUAKE FORUM	39
FIGURE 7: QUERY MONITORING AGENT INTERACTING WITH STUDENT	43
FIGURE 8: INFORMATION STRUCTURE AND STORAGE STRUCTURE	44
FIGURE 9: ENTITY RELATIONSHIP DIAGRAM	45
FIGURE 10: DEPLOYMENT WITH ONLY A CENTRAL DATABASE	48
FIGURE 11: EXAMPLE DEPLOYMENT WITH LOCAL DATABASES	49
FIGURE 12: LAYERS OF THE EQUAKE SYSTEM	50
FIGURE 13: EQUAKE USER INTERFACE	53
FIGURE 14: USER INTERFACE FOR SELECTING MESSAGE TYPE	54
FIGURE 15: SET UP FOR EVALUATION	57
FIGURE 16: THE TECHNOLOGY ACCEPTANCE MODEL (DAVIS, 1989)	59
FIGURE 17: RESULTS OF TEACHER EVALUATION	66

## List of tables

TABLE 1: OPEN SOURCE FORUMS	29
TABLE 2: MESSAGES POSTED SPLIT BY MESSAGE TYPES	60
TABLE 3: SUBSCRIPTIONS TO NOTIFICATIONS BY THE USERS	60
TABLE 4: INTERACTIONS BETWEEN MASSEY COURSE AND EIT COURSE USERS	61
TABLE 5: AVERAGE GROUPED VALUES FOR EVALUATION QUESTIONNAIRE	61
TABLE 6: HOW OFTEN USERS USED THE EQUAKE SYSTEM	62
TABLE 7: QUERY DISPLAY COMPONENT QUESTIONNAIRE RESULTS	67
TABLE 8: QUERY MONITORING AGENT QUESTIONNAIRE RESULTS	67

# 1 Introduction

To date, computers have been used in education for more than 40 years (Molnar, 1997). In particular tertiary educational institutions have increasingly employed computers as a tool for learning over the last 25 years in parallel with the evolution of the personal computers (PCs) (Freiberger & Swaine, 2000). Since around 1999 the use of intranet or Internet based e-learning systems has also been markedly on the rise within institutions (Paulsen, 2003).

Intranet or Internet based e-learning systems can support many administrative and learning tasks. Within the context of a course the tasks these systems are used for are: distributing course content, sending messages from the teacher to all students, enabling students to communicate with each other and for other learning activities (Avgeriou, Papasalouros, Retalis & Skordalakis, 2003). The type of software to support these tasks is usually called a Learning Management System (LMS), a Virtual Learning Environment (VLE) (Paulsen, 2003) or a Course Management System (CMS) (Mazza & Dimitrova, 2004). Such e-learning systems are currently being utilised within most New Zealand tertiary educational institutions and are increasingly being employed within more and more courses. An example is Massey University where in 2005 there were 2256 courses using their LMS solution (Browne, 2006). These tools have become increasingly important, especially for courses taught in distance education mode. They have largely replaced other mediums previously used for distance delivery such as mail, radio and television. This has presumably happened because Internet based e-learning systems increase the communication efficiency and provide new learning features.

These e-learning systems do however have some problems. The systems lack proper support for discourse-based learning styles, moreover they also lack features for collaboration between different courses and institutions. This thesis will show how current research tries to address these issues.

Described in this thesis is a research project aimed at improving these systems in relation to communication between students, using discussions as a way of learning and finding information in dynamic knowledge networks. It should enable students to engage in a student-centred community of learning across different institutions, share study and subject related problems, join study groups and receive expert advice. The focus of this thesis is thus, on the design and architecture of a system to support this.

This study is part of the project called eQuake (electronic Question and Answer Knowledge Environment). eQuake is aimed at designing, implementing and evaluating a web-based system for students in tertiary educational institutions within New Zealand. The system should, by design, also be applicable to any other (tertiary) educational settings, and some of the enhancements should also be usable for asynchronous computer based discussions in general.

The next sections of this chapter introduces the eQuake project in general, describes current discussion forum systems and identifies the shortcomings of these systems, particularly in the educational setting. A summary of the research questions and an overview of the rest of the thesis conclude this chapter.

## 1.1 eQuake project

The eQuake project is funded by the Tertiary Education Commission of New Zealand through their e-Learning Collaborative Development Fund; with its official title being *Agent based intelligent help system for New Zealand student community*. The project's primary objectives were identified as:

- a) better supporting discourse-based learning styles, as identified in priority groups such as Maori;
- b) providing wider student interaction across various institutions hence increasing exposure to multiple perspectives;
- c) reducing workload on teachers by reducing repetitive explanations and creating long-term archives of core student queries and solutions;

- d) alerting teachers to potential problem areas in students' understanding process.

This project was initiated by Massey University in cooperation with Eastern Institute of Technology (EIT), Auckland University of Technology (AUT) and Online Learning Systems.

The resulting system was developed on a model similar to asynchronous web-based discussion forums, since that was seen as the technology most easily extended to fulfil the project objectives.

## **1.2 Discussion forums: an introduction**

Internet discussion forums first appeared on the World Wide Web around 1995, one of the earliest systems available being Ultimate Bulletin Board (UBB) developed by Ted O'Neill (Kaiser, 2004). These systems are web applications that support asynchronous discussions online. They have similar functionality to dial-up bulletin board systems (BBS), news groups and mailing lists which had previously been the preferred tools of asynchronous communication over computer networks (Long & Baecker, 1997).

The main features of such a forum normally include the ability for users to post messages and replies to messages. Normally users are required to log in to identify themselves, although some forums also allow anonymous contributions. A feature that distinguishes different forums is how they visualise discussion threads (a message and its replies) in the user interface. Most forums today utilise either the linear or the threaded form for displaying messages or some combination of these (Newman, 2001). The linear form is when messages are displayed in a list; new messages are added either at the top or more commonly at the bottom of the list. Normally all messages are displayed in full. The threaded form on the other hand displays the messages as a tree, often only showing the titles for the messages not selected. Here new messages normally appear underneath and indented from the message they are replying to. Combined views are also possible that, for instance, have a tree of the message titles plus an area with the full content of the currently selected message. Some research has gone into alternative ways of visualising



the messages (Dave, Wattenberg & Muller, 2004; Donath, 2002; Donath, Karahalios & Viegas, 1999; Popolov, Callaghan & Luker, 2000); however these are mostly at the research level and are not widely used.

An additional feature that most forum solutions provide is to allow the users the option of receiving all new messages as emails when posted, this is for instance available in Moodle Forum (Moodle, 2006). This feature permits the forum to work almost like a mailing list and allows for the use of any email client to follow the discussions in a forum. These solutions do however still require the users to access the web page of the forum to post new messages themselves. An exception to this is Yahoo! Groups (Yahoo! Groups, 2006) and similar web pages, which allow posting both from email and the web, but this is because these systems are mailing lists with web interfaces and not proper forums. The main difference is that forums know the relationships between messages, while mailing list front-ends can only guess. Receiving all new messages as individual emails can lead to the users becoming overwhelmed. This is due to the high number of emails that can be generated each day by the high volume of discussion common in forums.

In newer forum solutions, including phpBB (phpBB, 2006), an alternative option lets users be notified by email only of replies to their own messages. This provides some of the benefits of mailing lists in that information is pushed out to users, but with a lower risk of overwhelming them.

The number of emails can also be reduced by providing a digest mode. This technique has been used by many mailing lists, for instance by sending out one email each day containing all new messages posted that day. But this only reduces the number of emails. Two problems still exist: all messages are still presented out of context and finding the ones that actually are of interest to the user might still be time consuming. This solution works better for mailing lists than for forums, as it is the norm for users to quote what they are answering or referring to in mailing lists. This means each message includes its context. In discussion forums this is normally not needed as the reply is shown directly under the message it is replying to when viewed with the web-

interface. However, mailing list digests still have the problem of requiring users to look through all new messages to find which ones are of interest to them. Digest emails alone do not solve the information overflow problem created by receiving forum messages as email.

Discussion forums can be valuable tools, but require refinements to help users manage the flow of information. This should be done both by improving user-interfaces and by providing better options for pushing the right information out to users.

### **1.3 Discussion forums in educational context**

Interaction among students has long been identified by educational research literature as a key success factor in learning (Brookfield, 1986; Slavin, 1983; Weller & Whitsed, 2004). Web-based discussion forums are now commonly used in tertiary education as tools to support this. These forums can be used as a supporting component alongside classroom lectures or as one of the main teaching tools in distance education. It has been demonstrated that having *learning communities* (Nijholt, 2002) and participating in discussion forums can be positive for several learning outcomes. These include: access to group knowledge, enabling a democratic learning environment where diverse contributions are supported; resulting in better user control over the learning process and increased motivation to learn (Bransford, Brown & Cocking, 2000; Muirhead, 2000). Transcripts from asynchronous online discussion systems have shown that those interactions had equal, or in some cases more, evidence of higher cognitive processes than face to face discussions (Heckman & Annabi, 2005). In particular it has also been shown that providing online discursive learning environments can encourage those students who normally rarely participate in classroom discussions to become more active (Pilkington, Bennett & Vaughan, 2000). Web-based discussion forums can facilitate learning, discourse and sharing of information among students no matter where they are physically located.

How teachers choose to use discussion forums within a course can differ quite widely. In some courses teachers post questions to start discussions but

otherwise try to stay in the background; in other courses, forums are mainly used by students asking questions to the teachers, and in yet other courses the forums are used for free discussions among students with no participation by teachers. A combination of these approaches can also be used. Which of these styles is the *best* is a question that cannot be easily answered and it might not make sense to try, seeing that each of the approaches will fit into different contexts. The choice of approach should probably be affected by factors such as the choice of pedagogy in the course and the learning styles and maturity of the students. A study by Mazzolini and Maddison (2003) showed somewhat surprisingly that increased participation in the forum by the teacher did not cause increased participation by the students. In particular the discussions were then generally of shorter length. However, the study also showed that students rated the instructors who participated frequently as both more enthusiastic and more knowledgeable. Discussion forums can be integrated in a course in many ways, how much the educator participates should depend on how they want the forum to be used.

Discussion forums are currently widely used in education. This has in many cases been successful, but how the forums are used within courses varies greatly and best practices still have to be established.

## **1.4 Limitations of discussion forums**

There are a few limitations to the utilisation of discussion forums. One drawback of all discussion forums in use today is that there is no way for the system to know the meaning of a user's message. To fully understand natural language is still an open problem in artificial intelligence. What we are requesting here is instead a way to know what the user is trying to achieve with posting the message. Is she or he posting a question, answering an existing question, making a suggestion, introducing a new idea or making a comment? Without this being explicitly defined by the user it would be a hard task for the system to identify all these types of contributions correctly. Knowing the intention would make it possible to find relationships between questions and answers, and to suggest existing answers when a new question is entered in the system. It would also be helpful for tracking how different

users use the forum. Do they mostly ask questions, mostly answer questions, mostly submit new ideas or mostly just read what is already there? It will also enable the system to keep track of whether a question is answered or not. All this should be possible without the system understanding the actual content of the messages, as long as the system knows the user's intention while posting the message.

Secondly, a forum is traditionally broken up into different topics or categories. The issue with this is that messages are frequently added in the wrong categories. This can be the case if someone does not know where their message fits in the categories already defined. The problem with this is that the categories lose their value as a tool to find information in a specific area. It can also mean that it takes longer to get a reply to a question as it might not be noticed by those that would know the answer, if they only follow *their* topic. In that case the message might have to be moved to another topic by a moderator before someone able to answer sees it. Keeping a topic structure useful as a forum grows in size can be a challenging task requiring much time from moderators.

Additionally, in discussion forums, many messages are repetitions of older messages. This commonly happens when a forum has grown to contain a sizable amount of messages. The users will then not know what questions are already answered in the system and will formulate a new question, which in turn has to be answered again, either in place or by pointing them to the existing answer in the archives. This also influences how effectively the forum can be searched, as these repeated messages can make it harder to find the answer the user is after in a result set. Repeated questions can deteriorate the quality of a discussion forum, because people quickly get tired of answering the same question again and again.

Getting email notifications of all new messages or of replies to the user's own messages is commonly available, but there are no other alternatives. For instance, notifications of replies to a specific message (by someone else) that a user finds of interest to them, of all messages within a specific topic or of all

messages by a specific user are not available. A higher degree of flexibility here would be an advantage for getting users the content they are interested in and limiting the likelihood of drowning them in too many messages.

While discussion forums are now commonly used, we believe there are many ways in which they can be improved further. This can be done for instance by allowing users to specify the intention of their messages and let the system use this information to manage the discussions better, or by incorporating a higher degree of freedom in how users specify which new messages they would like to be notified of.

## **1.5 Limitations of discussion forums in educational context**

Not all students are comfortable with the discussion forum medium. They might be facing difficulties with a number of things, for instance, not feeling at ease with the technology, finding it hard to use. Having a tracking system that enables identification of students with a low level of understanding or participation in the forum is therefore crucial. Current discussion forums can supply some tracking data based on user logins, messages viewed and messages posted, but lack the possibility to track what types of messages the users post. If a student posts many questions, this might be a sign that there is something that student does not understand. A simple form of tracking would then only register that the student has posted a high number of messages, indicating good participation, it would not detect that all the messages were questions which could indicate potential problems. Another example would be a student who is rarely posting, but only posting answers. A simple tracking would in that case identify the student as low in participation and potentially needing help, whereas the case might be that the student is an expert in the area. This could be further supported if his or her answers are highly rated by other students, which is also a factor a more complex tracking solution would be able to check. These extended types of tracking information could be used to automate both notifications to the teacher and to the students to ask if they want or need help. This tracking data could also be



presented to the teacher in a graphical way as previously done with the currently available tracking data (Mazza & Dimitrova, 2004).

Additionally, duplication or repeated questions can make the moderation of the forum more time-consuming than necessary for the teachers. Assuming a teacher-led style of discussion forum, this would cause the teachers to have to spend time answering questions that have been answered before. This in turn limits how the teachers use the forum, as they might not find time or desire to encourage student discussions by asking the students challenging questions. If a system can minimise the number of duplicated questions that have to be answered by a human, more time can be spent encouraging new discussions.

Courses are very much like separate islands in tertiary education. Students in different courses do not have access to each other's forums, even if the course content overlaps and similar questions might come up in both courses. This is true both inside an institution and between different institutions offering similar degrees or courses. Allowing forums to be shared between several courses would increase the knowledge base available and expose students to multiple perspectives.

Reusing parts of the knowledge base accumulated previous times a course has been run is often desirable in education. This is not supported in forum systems, as they normally only give the options to keep everything or remove everything. What is desirable is for instance being able to keep some high quality discussions or messages answering key questions, these could be identified based on ratings or number of times viewed. It would also be desirable to then be able to hide the user identities of those users who originally contributed these messages. A partial reuse of existing messages like this might be very useful in some settings, though the mechanisms for message selection would be a key factor.

Several issues for improving discussion forums in education have been identified. Better tracking solutions based on systems with knowledge of message types would give teachers more insight as to what is going on with

the students. Having forums shared by multiple courses would allow more knowledge resources to be connected.

## **1.6 Research questions**

The key research questions analysed in this thesis are:

1. Can a distributed discussion system be developed that enables sharing between multiple tertiary educational institutions?
2. Can such a system be integrated with existing Learning Management Systems?
3. Would students find the features of the new system useful?
  - 3.1. Would students find the system's automatic answer finding behaviour easy to use?
  - 3.2. Would students find the system's extended notification features easy to use?
4. Would teachers find the system useful and be willing to integrate this type of system in their courses?

## **1.7 Thesis structure**

The structure of this thesis is as follows: the next chapter gives an overview of similar systems to the one proposed in this study and their use in educational settings. The third chapter describes the requirements identified for the new system and the fourth explains the development environment chosen. The fifth chapter illustrates the architecture and the implementation of the developed system. The sixth chapter discusses some specific design and implementation decisions in detail, while the seventh chapter contains the testing and evaluation. The final chapter reports on the conclusions drawn from this study and on ways of extending this work.

## 2 Literature review

This chapter looks at various discussion forums and asynchronous discursive learning environments as described in literature. Its primary focus is on features that would be of benefit for the goals of the eQuake project.

### 2.1 Knowledge Forum (CSILE)

Within the field of discourse based learning, one of the most prominent systems is CSILE (Computer-Supported Intentional Learning Environment). The later versions of this system are known as Knowledge Forum, principally developed by Scardamalia and Bereiter (1993, 1994 and 1996). This system was built to support intentional learning and has clear socio-cognitive underpinnings. Much learning occurs unintentionally, but for useful school learning Scardamalia and Bereiter (1994) argue that intentional learning is often required. Socio-cognitive theories state that knowledge seldom appears *out of nothing*, more commonly it appears through the interactions of the ideas of several individuals (Hjørland, 2002). The system supports this by allowing users to specify their theories, which other users can then build on. An additional key point of this system and the associated theory of learning is that an important part of inquiry is for the learners to specify what they need to learn, or what they do not know. The system tries to support this by creating an environment similar to the scientific environment of researchers. In the system *notes* can be posted, first in *draft* form that students can work on and later in *candidate for publication* form which requires peer review by other learners before it is *published* (Scardamalia & Bereiter, 1996). Referring to others notes and building on them is encouraged. Linking several notes together to gain an overview of a subject is also a central feature. The early versions of Knowledge Forum (CSILE) were pioneering efforts in the field of computer supported collaborative learning (CSCL).

Knowledge Forum is not a traditional discussion forum system. It is built specially for the task of learning, but it has good support for carrying out online discussions. This as the system supports comments to the different types of notes that can be entered in the system, and as it supports notifying



the author of a note when someone posts a comment to that note (Scardamalia & Bereiter, 1996).

When examining their later works (Scardamalia, 2002; Scardamalia & Bereiter, 2005; Scardamalia & Bereiter, 2006) it is apparent that evaluating Knowledge Forum is impossible without looking at the underlying educational theory of Knowledge Building. This theory can be classified as a socio-cognitive theory which emphasises that students formulate ideas and that ideas, not activities drive what happens in the classroom (Scardamalia & Bereiter, 2005). One distinct feature of this approach is thereby to push some of the meta-learning decisions to the students instead of being handled by the teacher. This happens by students having to formulate what they do not know and thereby what they need to learn more about, to better understand a specific problem (Scardamalia, 2002). The foundation of intentional learning is that students actively pursue knowledge.

When Scardamalia and Bereiter (2006) compared Knowledge Forum with traditional threaded discussion forums, these main differences were found: Knowledge Forums consists of *notes* and *views*; a note is comparable to what is normally called a message in other forums, with the exception that notes can always be edited or updated by their authors. Notes can also link to any other notes, and proper source referencing will then be automatically inserted. Views link different notes together and can be freely defined by the users, unlike in a traditional forum where the only options for listing messages are normally *threaded* form or chronological lists. In some sense Knowledge Forum can be said to have some traits typical of a Wiki (Wikipedia, 2006b) or a collaborative authoring environment, in that the main focus is on building the knowledge of the community. Although in Wikis the actual content often does not show who contributed it, while Knowledge Forum keeps the author identities visible. This allows for easier representation of multiple (conflicting) points of view, while still having powerful tools for discussing, summarising or further analysing existing notes. Knowledge Forum has key advantages when used for Knowledge Building as compared to more commonly used web forum systems.

Knowledge Forum was first aimed at primary school education in the 10 to 14 age group, but has since been used at all levels from kindergarten to university courses. Numerous studies have shown that the use of this system and pedagogy, which are normally seen as a more or less integrated package, has in most cases had clear positive outcomes for students learning. Both for their meta-reflective skills of knowing what they know, and what that means to them, and also in standardised tests measuring set curriculum goals. It has although been recognised that for both teachers and students it can be a challenge to learn to work according to this new pedagogy. (Bielaczyc & Blake, 2006; Caswell & Bielaczyc, 2002; Rahikainen, Järvelä & Salovaara, 2000; Schacter, 1999; Winter & McGhie-Richmond, 2005)

## 2.2 Future Learning Environment (FLE)

The Future Learning Environment (FLE) is an open source web based Computer Supported Collaborative Learning (CSCL) environment written in the Python language (Leinonen, Kligyte, Toikkanen, Pietarila & Dean, 2003). It is based on the theory of Knowledge Building as formulated by Scardamalia and Bereiter (1993) and extended as the Progressive Enquiry model by Hakkarainen, Lipponen and Järvelä (2002). FLE contains three different components for students to use, namely:

- *Webtop* which is a personal space where the users can store files, links and knowledge building notes.
- *Knowledge Building* which is a group discussion environment for structured knowledge building where each discussion note is labelled with its thinking type.
- *Jamming* which is an environment for collaborating on developing any digital artefacts, these can be pictures, sounds, videos, texts, etc. The tool encourages exploring different design possibilities by allowing multiple versions to be created of an artefact.

(Leinonen et al., 2003)

The *Knowledge Building* part of this system is the part most related to the eQuake project. This part is similar to Knowledge Forum, but without the features for users to create *views*.

## 2.3 Student Assistant Agent in FLE

An agent system called Student Assistant Agent has been implemented (Mørch, Dolonen & Omdahl, 2003) extending FLE. This system monitors the students' participation in the Knowledge Building and can make suggestions to a student based on both the student's actions and the state of the discussion. These suggestions are all on the meta-discussion level, in that their goal is to advise the students on recommended interaction techniques. The Agent has no concept of the actual topic being discussed. These suggestions can for instance be:

- a suggestion to reply to a specific note if no one else has,
- encouragement to contribute if the user is less active than the other users,
- a notification informing that the user has contributed considerably more than the others and might be dominating the discussion too much.

(Mørch et al., 2003)

A further enhancement was the development of a RuleEditor for use by teachers to customise when and what kind of feedback the Student Assistant would give (Mørch, Dolonen, & Nævdal, 2006). This is an important addition as it is very hard to estimate with what frequency and which notifications will be helpful at system design time. These factors would also likely be different for various usages of the system, requiring this level of teacher customisation.

## 2.4 Discourse environment for Shadow netWorkspace

Jonassen and Remidez (2005) have developed a structured discussion forum or discourse environment for Shadow netWorkspace (SNS) (Shadow netWorkspace, 2006). The most interesting feature of this implementation is how they have limited the available *thinking types* based on the context of the message being replied to by the user. Figure 1 shows which options are available at the different stages of a discussion. When starting a new discussion only the Problem level thinking type is available, so each discussion will always be started with a Problem statement. A Problem statement can in turn only be replied to by Solution proposals, while Solution

proposals can be replied to by any thinking type from the Warrant level and Warrant level messages can only be replied to by Evidence level messages.

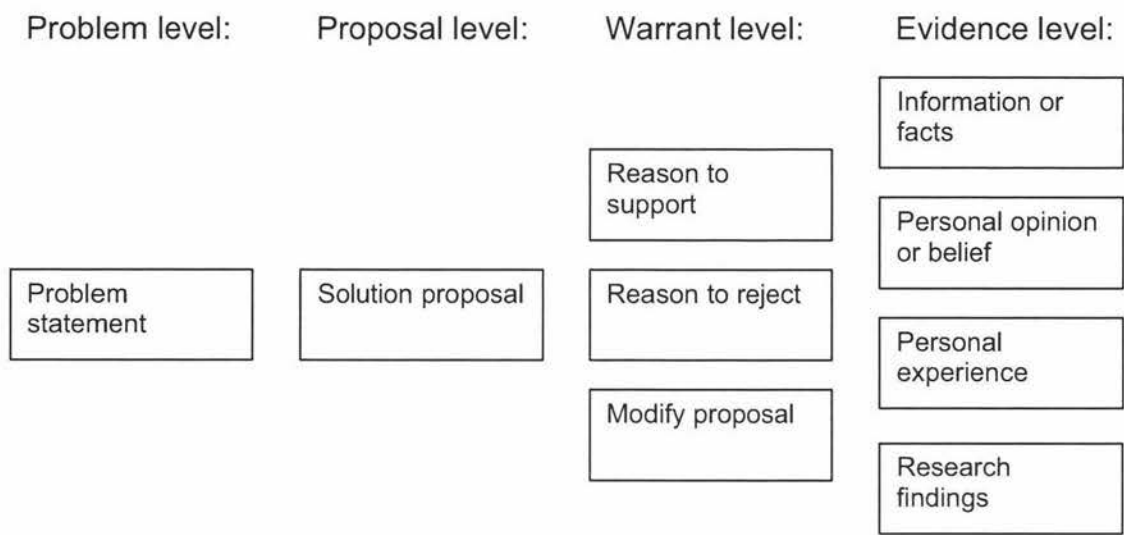


Figure 1: Thinking types for SNS discourse environment

A further characteristic of this system is, to allow the teacher to set up their own set of thinking types, and also to specify what the forms for adding new messages of each of these types should include. A teacher can for example add a thinking type where the message form includes some dropdown boxes or radio buttons to select among some options in addition to the standard text area.

In their evaluation of this system with 21 graduate students Jonassen and Remidez (2005) found that the system had worked mostly as planned, but did not show proof of having been effectively used to conduct an argument. For the evaluation, the researchers had predefined one problem to be discussed and limited the students’ access to thinking types of the Proposal level and onwards. The results of their study show that the *Information or facts* and *Research findings* thinking types had not been used at all. When the researchers recoded each message using the same thinking types as the students had used they found that the students had a tendency to use the *Personal opinion or belief* thinking type also for messages that were of other types. Another observation was that *Modify proposal* messages often were compound messages also containing complements or critique of the existing

proposal. The researchers suggest that, to get students to effectively conduct arguments using the system, it would be required to give them an introduction on how to use constraint based discussion forums to support argumentation before using the system (Jonassen & Remidez, 2005).

Another option is perhaps changing some of the message categories or loosening restrictions on sequencing. For instance the *Modify proposal* category, that was frequently misused, could be extended to consist of both a text area for agreeing or disagreeing to the previous proposal and an area for specifying the modifications to the proposal. In effect this would add two *messages* in the system from one user action. Limiting the available message types, based on the type of the message being replied to, is an interesting approach.

## 2.5 ForumPlus

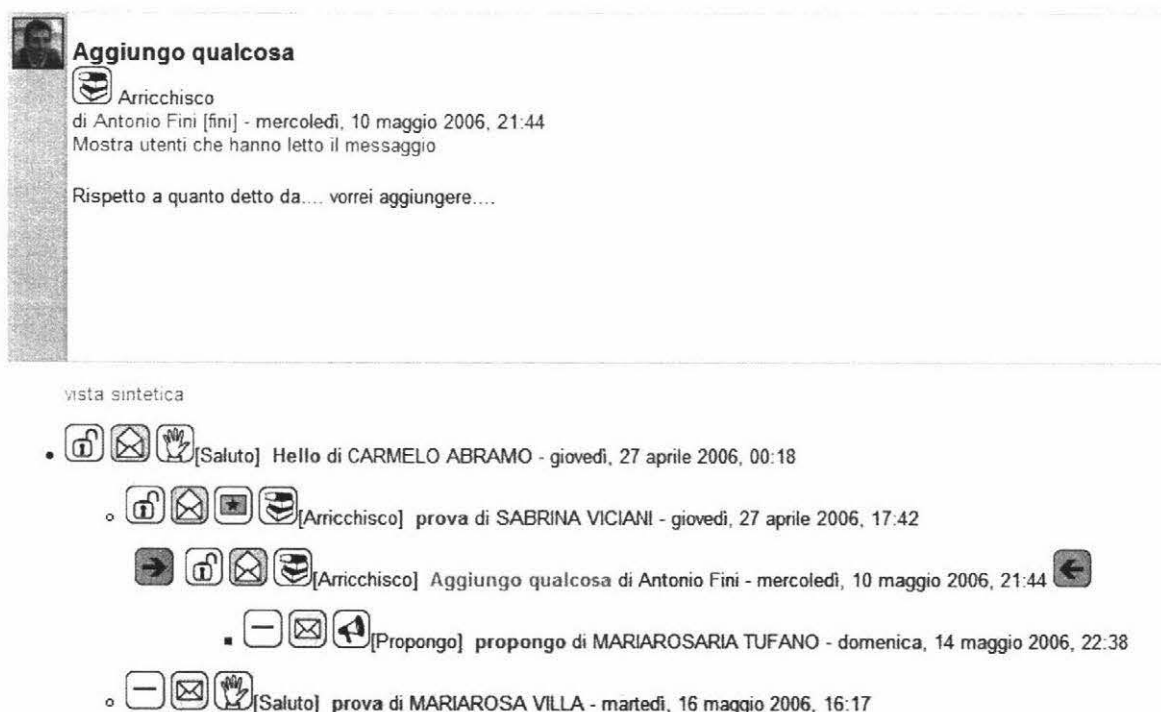


Figure 2: ForumPlus user interface (Fini, 2006)

ForumPlus as implemented by Calvani, Fini, Pettenati, Masseti, and Sarti (2005) is an extension to the forum module of the Moodle LMS. This forum was developed with the goal of integrating CSCL theories within open source

learning management systems. The forum differs from the standard Moodle forum in several ways. Students have to specify the *ThinkingType* of each message they enter in the system, these types are defined by the teachers when setting up the forum. Another difference is how messages are visualised in the user interface. In ForumPlus the active message is displayed on the top of the screen, while a tree of message subjects are displayed underneath as seen in figure 2 (Calvani et al. 2005).

## 2.6 Flash forums and ForumReader

The ForumReader, developed by Dave et al. (2005) is not used for education, but is still an interesting system to observe. They class this system in a special type of discussion forums they call *flash forums*. What identifies a flash forum is that large numbers of messages are submitted during a limited period of time. The messages also usually only form short threads and authors identities are less prominent. Another example they use is the comment section for each news item at the Slashdot website (Slashdot, 2006). Flash forums have also been known as Massively Parallel Conferences, large-scale distributed meetings or as discussions in a virtual public (Dave et al., 2005). For users a flash forum can be challenging to navigate, as there are so many messages on one topic and they are not extensively threaded or split up in sub-discussions. A common way of helping the users find which messages other users have found to be of greater value to them, is by using rating systems. These systems allow users to give messages posted by other users a rating. The rating can be based on how relevant they find the message to the discussion, how insightful, how funny or any other scale suited for the context. The Slashdot website is a famous example of this, but the feature is currently getting more common in other places online as well, for instance the OSNews site (OSNews, 2006). An additional interesting (and challenging) aspect of flash forums is how to visualise discussions with large numbers of related messages. In ForumReader a thumbnail visualisation is used that shows all messages in a discussion as small boxes. The boxes reflect the length of the messages and the placement of the boxes signify where the messages fits into discussion treads. In addition there is a text view, which displays the message in full, used for the actual reading. Features to search the discussion



by various criteria, that then highlight the matching messages in the thumbnail view, are also provided (Dave et al., 2005). Although forums for learning are different from flash forums, some of the issues they face with scalability can be transferred. Especially for a system like eQuake that needs to scale to larger numbers of messages than traditional discussion forums used in education.

## **2.7 Dynamic Frequently Asked Questions environment (DFAQ)**

A Dynamic Frequently Asked Questions environment (DFAQ) developed by Ng'ambi and Hardman (2004) is an environment that enables the students to build a knowledge resource by asking questions and adding (possible) answers. The users in the system are all anonymous; this was done to create a *low risk* environment where no one would feel stupid for asking their question or submitting their possible answer (Ng'ambi & Hardman, 2004). Letting students answer each others questions has the big advantage that, as they are all currently learning the subject, they are often better at describing solutions that other students understand. This can at times be better than an answer from a teacher who normally will have a considerably higher level of knowledge within the area, and might answer on a different level of complexity. Teachers can intervene in the system when they find questions that remain unanswered or answers that are incorrect. Learners and teacher also have the option of rating responses as *very good*, *good*, *poor* or *very poor*. Having a database of what students find hard to learn helps the teacher find what areas might need more attention in later lectures or activities (Ng'ambi & Hardman, 2004).

The DFAQ environment was tested by researchers in a university course by having postgraduate students read two challenging academic texts and then post questions and answers based on these. An important goal of this exercise was also to encourage critical questioning behaviour in relation to reading academic texts, as that was considered an important skill to acquire for the students. Based on looking at the questions entered in the system and on interviews conducted with the students, critical questioning skills seemed to

be improved while students found the system helpful (Ng'ambi & Hardman, 2004).

DFAQ is interesting in that it focuses more on asking and answering questions than on discussions. Another important issue that can be raised from this study is how showing or hiding the users' identities influences how they interact online. Users of DFAQ indicated that they liked the anonymity it provided (Ng'ambi & Hardman, 2004), while other successful systems like Experts Exchange (Experts Exchange, 2006) are based on encouraging users having an *online presence*. This is likely to be a cultural issue, so what works best in education might depend on the student population in question.

## **2.8 Finding knowledge resources**

Finding the correct information when it is needed is in many circumstances a critical task. Organisational memory and collaborative help systems can be considered central examples in this regard. Systems like Answer Garden and Answer Garden 2 (Ackerman & McDonald, 1996) were early examples of computer systems developed specifically for this. These systems helped users both by finding information directly and, if that failed, by identifying experts able to help them. Further systems have been developed using intelligent agents technology to match novices and experts based on their knowledge profiles (Vivacqua & Lieberman, 2000). Systems matching users based on Latent Semantic Indexing (LSI) of the textual documents that can be related to the users have also been developed with the aim of further improving the matching accuracy (Reichling, Schubert & Wulf, 2005). This way of finding knowledge resources, both in stored messages and as other users, is an important usage scenario for forums in an educational setting, and thereby the eQuake system in particular.

## **2.9 Social creativity and social capital**

An important point of encouraging students to cooperate is the concept of social creativity. As defined by Fischer (2000) this states that interacting members of a group can come up with better solutions to some types of problems than any one of the participants could have come up with on their



own. When creating groups or networks for this purpose it is important to take pre-existing social networks into account as that will affect how well each participant performs (Cho, Lee, Stefanone & Gay, 2005). When building computer systems to support this, it is also important to be aware of and design for the social capital within the group. Social capital can characterise properties of a group such as trust, networks and norms (Fischer, Scharff & Ye, 2004).

The Experts Exchange website (Experts Exchange, 2006) is an example of a web forum that has been quite successful in this regard. It contains a market of points for answering questions, where members increase their amount of points by answering other members' questions. Status within the community is then reflected by including a listing of the members with the highest amount of points. A similar approach with an internal market for question priorities for use within an educational setting has been suggested by Tretiakov, Smestad and Kinshuk (2006), proposing this will encourage students to be more constructive in their discussions and more helpful in answering each other's questions. This has not yet been fully implemented and evaluated, but could possibly be implemented as an extension to the eQuake system.

The next chapter will describe the requirements determined for the new system based on this literature review and other input.

### 3 Requirements analysis

This chapter begins by giving an overview of the requirements that have been gathered for the new system. This is followed by more details and explanations for some of the specific requirements.

#### 3.1 Requirements identified for new forum system

Requirements for the proposed new discussion forum system have been gathered. These requirements are based on discussions among the eQuake project group at large, consisting of the students responsible for the implementation and teachers from the tertiary institutions involved. This in addition to the analysis of the properties of existing (general purpose) discussion forum systems for use within an educational context reported on in chapter one. And also the examination of systems enhanced in similar education specific ways as what this research study is targeting in chapter two. The requirements identified can be summarised as follows:

- Based on the assumption that being presented with multiple perspectives and sharing knowledge across larger groups is beneficial for students, one forum should be shared by multiple courses. This should be possible both inside one institution and across different institutions (using different learning management systems). This is in tune with other efforts to define Learning Objects (Friesen, Fisher & Roberts, 2004) for sharing among institutions and building repositories of reusable learning content (Baraniuk, 2006) for easing access to knowledge.
- Users have to specify the *thinking type* or the type of content in their messages, like *question*, *answer*, *comment* or *idea*. This is so that messages can be handled in a more intelligent way in the system, and to make the users more aware of the intention of their message. Since the system will use the message types, these have to be defined when the system is developed, and not by the teachers when using the system.
- New questions should be intercepted by the system, and the existing knowledge base should be searched and a list of possible answers

displayed before the users' questions are added to the system. This is to allow the users to quickly find answers to their questions rather than having to browse or search through the entire forum to see if the information already exists. This feature should also attempt to avoid the creation of duplicate messages in the system.

- Users should be able to subscribe to get notifications when new messages are posted. They should be able to subscribe to forums, to topics or to replies to specific messages. For messages, they should also have the option of specifying if they want notifications of all replies or of only answers. Notifications are an important way of pushing information out to the users, especially since users have indicated that this information will be of interest to them.
- Notifications should at least be available to the users as emails. Other mediums like text messages to mobile phones and personalised RSS (Really Simple Syndication) feeds would also be of interest. Choice of medium could then be done either as a user preference or based on the assumed importance of the new message for each user.
- Students should be able to rate how helpful a message was for them. This allows highly rated messages to be given a higher ranking in search results and marked as *highly rated* within the user interface.
- The student usage of the system should be monitored, and teachers and students notified on detection of low levels or specific patterns of usage, such as a student only asking questions.

We consider a system implementing these features to be a substantial improvement to the systems currently used within education. As it should better support discourse based learning styles, while also providing an environment that better helps finding information.

### **3.2 Message types / thinking types**

Forcing the users to specify the type of message they are posting is one of the biggest differences between eQuake forums and standard discussion forums. There are two reasons for introducing this:

1. The main reason is to make it possible for the intelligent agents in the system to reason about the messages.

2. The other goal of specifying message types is to encourage users to be more aware of what kind of contribution to the discussion they are trying to make.

If this had not been done, the system would need to use a full artificial intelligence (AI) solution to analyse messages to try to identify these types. The types are needed by the enhanced features of eQuake for intelligently tracking discussions and finding answers. Using an AI approach to solve this problem would increase the complexity of the system significantly and might not even be possible with the current state of the art in natural language processing. So, by forcing the users to specify the type of message, the system can skip directly to the next stage where it tries to find answers to questions and does other similar tasks since the messages are already classified. This approach utilises the advantages of both parties (Grieser & Lange, 2005); the users can easily specify the type of message they are writing, while the system is efficient at searching large amounts of data for similarities, thereby solving a complex task with a relatively simple solution.

### **3.3 Selection of message types**

A set of message types had to be selected for the system. While some systems evaluated (Calvani et al., 2005; Jonassen & Remidez, 2005) allow teachers to specify message types when running a forum, this had to be decided at system design time for eQuake. This is because the message types in eQuake are not only to help the students structure their discussions, but also to help the system make sense of the relationships between the messages. The goal when selecting message types was thus to strike a balance between how many types would be needed by the system to be able to reason about the messages while not overwhelming (and annoying) the users by requiring them to choose from an extensive list. A number of classification schemes were evaluated (Bodzin & Park, 2000; Dringus & Ellis, 2004; Jonassen & Remidez, 2005; Watson & Prestridge, 2003), and in the end the following categorisation was selected:

- Answer
- Comment
- Information gathering question
- Solution seeking question

- Suggestion/Idea

Tutors also have the possibility to add:

- FAQ question
- FAQ answer

This classification is assumed to strike a balance, giving detailed enough information for the message analysing components of the system and making sense for users without requiring extensive training.

### 3.4 How students use discussion forums

Students use discussion forums to ask questions or to find information in various ways. We here propose to split this usage in three types of scenarios and then look at how the eQuake system will support each of them:

1. Asking course specific questions, for instance “When is assignment 1 due?” this type of questions relate directly to the course, but not to the content.
2. The second type of use is information gathering, trying to find information about something the user knows little about, for instance “What is PHP used for?”.
3. The third way is to discuss a topic; this is when the users try to validate, extend and cement their existing knowledge, for example, “Is `mysql_connect('localhost', 'mysql_user', 'mysql_password');` the only/best way I can connect to a database in PHP, or are there other options?”.

For the course related type of questions (1), eQuake can help by enabling more means of tracking each message to check if it has been responded to (by a tutor). The system’s ability to search for similarities between new questions and existing messages should also be useful for this type of questions. But this usage is not what has been the main focus for eQuake, and to support it fully within eQuake would most likely require the system to allow for forums or topics private to one course in addition to the global forums shared by multiple courses. An issue here is that students would need to be trained in what to post where. Having one forum for course specific questions and one forum for topic related questions might still be a difficult separation to implement, since a discussion can easily cross over from one category to the

other. Another approach might be to keep everything in one forum, but make the eQuake system more aware of the concept of courses or classes of students. Thereby making it responsible for ensuring that course specific messages are easier to find for the students they are relevant to and less likely to be found by students they are irrelevant for.

The second, information gathering, type of usage is where the larger knowledge base of the eQuake system is the most helpful; by connecting more users and more messages in inter institutional forums. The system's search of similarities between messages is also important here, as the larger pool of knowledge requires better searching capabilities for finding the correct piece of information.

Finally, using the system in a discussion oriented manner should be helped by the explicit specification of message types. This as it hopefully will encourage students to reflect and be more aware of what kind of contribution they are trying to make to the discussion. More precise message classification schemes might also be desirable for this type of usage though. The added number of users should also be an advantage in this case, since this would give access to more diverse perspectives.

Some functions of eQuake like the monitoring of students and the great flexibility in subscriptions to notifications can be considered as advantages for all these uses of discussion forums by students.

Discussion forums can be used for many different ways of learning related questioning by students. This is an important point to be aware of when designing systems to support these different scenarios. We have hopefully shown here that the features of eQuake can be beneficial to all these, though the main focus of the development have been to support the topic related questions (2) and learning by discussing (3).



### 3.5 Integration with learning management systems

A key success factor in this project is for the developed system to integrate properly with the existing learning management systems that are used in the different courses that will use the new system. An examination of open source environments, including Moodle (Moodle, 2006), Sakai (Sakai, 2006) and Claroline (Claroline, 2006) was undertaken. Since Moodle is the LMS most commonly used in New Zealand of the ones in this list, and as it has a quite clear interface for making plug-in extensions or modules for the system, Moodle was selected as the primary LMS to integrate with.

Since integration with closed source systems might also be required for the success of this project, WebCT (WebCT, 2006) and BlackBoard (BlackBoard, 2006) were evaluated. They are both in use within major New Zealand institutions. The main area that was examined with these systems is how they allow for external developers to integrate with their system by writing plug-ins or extensions.

Plug-ins for BlackBoard are called *Building Blocks*. A Software Development Kit (SDK) that supports developing these for newer versions of BlackBoard (6.0 – 6.3) is freely available. *Building Blocks* are normally developed as Java Web Applications (using JSP or Servlets), but .NET development is also possible.

For WebCT Vista, the newer version of WebCT, plug-ins are called *PowerLinks*. The company charges money for the SDK that supports development using Java or Web Services. For older versions of WebCT, such as Campus Edition 4.1 that is currently in use at Massey University, the SDK is available free to license holders and the development environment is Perl.

In view of the fact that the development team was located at Massey University and had no access to BlackBoard servers, it was decided to try to integrate with WebCT as the secondary LMS for the development.

The two learning management systems selected, Moodle and WebCT, should complement each other in proving that the system is flexible, by interacting with both open and closed solutions. They should also be usable as reference implementations for creating further plug-ins to interact with other learning management systems.

The next chapter will outline the development plan, describing technologies used and how the development team cooperated.



## 4 Development planning

This chapter will present various technologies selected for use within the system. Following on from this, the chapter gives a high level overview of the proposed system before describing the development approach and environment chosen for the implementation. Finally it presents how tasks were split up for the implementation, and how the development team communicated.

### 4.1 Technologies used in the system

This section introduces various technologies, before explaining how they are proposed to be used in the system in the following sections.

#### 4.1.1 Web Services

The Web Services concept has recently gained a lot of popularity as part of the Service-Oriented Architecture (SOA) approach. SOA is often defined by nodes making resources available to other nodes as independent services. The important properties of the services are usually defined as being loosely coupled, with well specified interfaces and not exposing their underlying implementation details (Kobielus & Manes, 2004).

Web Services, as defined by the W3C (W3C, 2006), are based on Extensible Markup Language (XML) and normally implemented using Web Services Description Language (WSDL) definitions to describe each service and using SOAP envelopes for sending requests and responses between the nodes.

#### 4.1.2 Agents

The term *agent* has been used to signify numerous things within the field of computer science (Wooldridge, 1999), and an exact definition is therefore not possible. The reason for using this term for various components of the proposed system is that they are components acting and interacting on behalf of (representing) a user to gather and return information for that user. More on educational agents and why they are used within the eQuake system can be found in Zhang, Kinshuk, Smestad, Yang, and Jeffery (2006).

### 4.1.3 Object relational mapping (ORM) tools

Two commonly used ORM tools for development of web applications are *Hibernate* and the *Active Record* part of the *Ruby on Rails* framework (Peak, 2005). *Hibernate* (Hibernate, 2006) is an object/relational persistence and query service for Java, while *Rails* is an integrated web application framework written in the Ruby language. Basically these tools automatically map data structures between a relational model stored in a database and an object model as used for object oriented programming. They are then in charge of updating the database when changes occur to data in the object model.

These technologies were seen as key to the successful implementation of a system meeting the specified requirements within the timeframe allocated.

## 4.2 Survey of existing open source forum systems

A large number of the features planned were basically enhancements to current discussion forum concepts. Existing open source discussion forums were therefore examined to see if it would be preferable to start with the code base of one of them and build on that, instead of starting development from scratch. Table 1 shows an overview of the forums evaluated in this study.

Table 1: Open source forums

Name	URL	License	Programming language	Message structure
Beehive Forum	<a href="http://beehiveforum.net">http://beehiveforum.net</a>	GPL	PHP	Threaded
phpBB	<a href="http://www.phpbb.com">http://www.phpbb.com</a>	GPL	PHP	Flat
PunBB	<a href="http://www.punbb.org">http://www.punbb.org</a>	GPL	PHP	Flat
JavaBB	<a href="http://www.javabb.org">http://www.javabb.org</a>	Apache License	Java	Flat
JForum	<a href="http://www.jforum.net">http://www.jforum.net</a>	BSD License	Java	Flat
Yazd	<a href="http://www.forumsoftware.ca">http://www.forumsoftware.ca</a>	Apache License	Java	Flat
Moodle Forum	<a href="http://moodle.org">http://moodle.org</a>	GPL	PHP	Threaded / Flat
Forum Plus	<a href="http://moodle.org/mod/data/view.php?d=13&amp;rid=354">http://moodle.org/mod/data/view.php?d=13&amp;rid=354</a>	GPL	PHP	Threaded

The preferred programming language would be Java, because that was the language selected by the development team for use in the backend of the system. The backend part would be implemented from scratch, so reuse was only possible for the Presentation Layer of the system.

For the message structure, support for threaded messages was preferred as a data structure storing this information would be required to fulfil the requirements of the new system and this form of visualisation was preferred by the user interface developers.

Three different open source licenses were in use by the projects evaluated:

- GPL – GNU General Public License (Free Software Foundation, 1991)
- BSD – (Modified) Berkeley Software Distribution License (XFree86, 1996)
- Apache License (Apache, 2004)

The main difference between these licenses is that with the GPL any modifications or extensions made to the source code also have to be open source (under a compatible license). The other licenses are less strict in this sense and can easily be combined with closed source code. Hence for the purposes of this project, we assumed that our code could be made freely available, as that would not negatively affect the funding agency. Therefore, we should be able to interoperate with any of these licenses.

Evaluations of other optional features were also undertaken. These included evaluations as to how well and in what language (some were not English) the source code was documented. Additionally, assessments of how well their data models would match with the additions required for the proposed new system were also carried out.

In the end it was decided to base the system on the existing Moodle Forum module. This because it already was decided to use Moodle as the main LMS to integrate with, and the source code seemed the best suited. Since Web Services were intended to be used for communicating between the system's layers, the fact that this would require the Presentation Layer to be written in

the PHP language instead of Java was not seen as a mayor problem. Using Web Services would hide the implementation languages of the various layers from each other anyway.

### 4.3 Overview of proposed system

A high level overview of the main components and main flows of control of the proposed system can be seen in figure 3. The diagram shows how the system is split in three layers and also how it integrates with the existing LMS in use. The full details of the system will be presented in the next chapter.

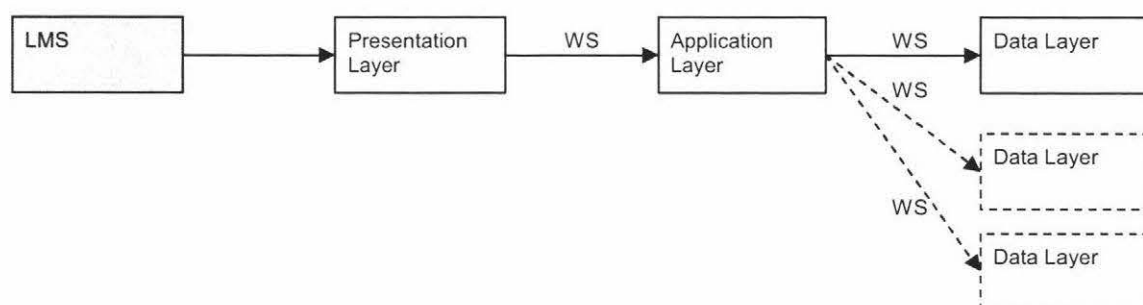


Figure 3: High level system overview

Figure 3 also shows where Web Services are used in the system, with the lines marked WS. This occurs in two different ways. Firstly, a request from the user interface implemented in the Presentation Layer is wrapped in a SOAP envelope and then a Web Service call is made to the main Application Layer. These two parts of the system can be located on the same physical server or on different servers. The second use of Web Services in the system, which is the most distinctive feature of the high level architecture, is for making requests from the Application Layer to the Data Layer. In this case, one request from the Application Layer will normally call both the Data Layer Web Service of this instance of the system and the Data Layers of all other institutions participating in this discussion forum. This is what the dashed lines and additional Data Layers in the figure represents, how many Data Layers on different servers will be called then depends on how many institutions are participating in the forum.

## 4.4 Development approach

Java was chosen as the programming language for the development of the Application Layer and the Data Layer of this system (these were to be implemented from scratch). This was because the team members had previous experience with development using this language. Moreover it was assumed that a *strict*, statically typed, language like Java would better encourage good design and coding standards. Use of interfaces to define how different parts of the system would interact with each other was also seen as important. This was both for describing the functionality required in the different components before starting the implementation, and also as a means to ensure that components developed by different developers would interact correctly. Java was therefore considered the logical choice for the backend of the system.

To develop the Presentation Layer, or user interface part, it was decided to use the PHP scripting language and extend the existing forum module of Moodle (which is developed in PHP) after evaluating various open source forums as discussed earlier. By starting with an existing user interface it was quicker to reach the stage where a running prototype of the system could be tested, and further enhancements could then be added iteratively. As Moodle would be an important LMS to interact with and since their forum module seemed a good starting point for the planned extensions, this approach was chosen. Having the user interface implemented in another language than the Application and Data Layers of the system was not seen as a negative point, but as a positive, as it enforces using the proper abstractions between the layers of the application. These abstraction layers were implemented using Web Services, which are language neutral in that they do not expose the language used in the implementation of the services they provide. Using different languages for different layers of the system was therefore not a problem. Building on the forum module of Moodle would also allow reuse of other existing Moodle components if required. Currently the user handling and the support for themes in Moodle are used. Basing the Presentation Layer on extending the existing Moodle Forum module gave this component a head start in development.

To develop the Data Layer a modelling approach was followed by using a modelling tool to design an entity relationship (ER) diagram. Originally, the design was based on the data structure of the Moodle forum module, but later it was changed and extended considerably to support the specific requirements of the new system. That diagram was then used to generate an SQL script to create the actual database. The database was in turn accessed to generate Java persistency classes for use with Hibernate in the Data Layer. Choosing a modelling approach for the Data Layer ensured that the documenting diagram was up to date, as the code was generated from the diagram.

A code driven approach was selected for the Web Services implementation, in that the Java classes implementing the functionality were written first. The XML based WSDL descriptions and supporting files were then generated from the Java source code, exposing the public methods of the classes as Web Services. This was done using the Eclipse Web Tools Platform (WTP) (Eclipse Web Tools Platform, 2006) package and the files were afterwards tweaked manually to correct some minor errors in the generation. In operating the Web Services, Apache Axis (Apache Axis, 2006) is used in combination with Apache Tomcat (Apache Tomcat, 2006). Web Services were integrated in this way to allow the developers to focus on the functionality of the system, not on the specifics of Web Services technology.

For this project a range of development approaches were used for the various components of the system. This was done as each approach was considered to give substantial advantages for the component in question. A single unified approach would not have been as effective for each of the different cases.

## **4.5 Development environment**

Free software options were chosen when available for the development environment. With the exceptions of the Sun JDK (Java Development Kit)<sup>1</sup> (Sun JDK, 2006), Microsoft Windows (Microsoft Windows, 2006) on the

---

<sup>1</sup> Sun is currently in the process releasing this as open source (Sun, 2006).



developers' workstations and Microsoft Office (Microsoft Office, 2006) for some diagrams and presentations, all software used was open source. Eclipse (Eclipse, 2006) was used as the Integrated Development Environment (IDE), with plug-ins for web development (Eclipse Web Tools Platform – WTP) (Eclipse Web Tools Platform, 2006), PHP development (PHPeclipse) (PHPeclipse, 2006), Hibernate (Hibernate, 2006) object-persistency with MiddlegenIDE (MiddlegenIDE, 2006) and Subversion connection (Subclipse) (Subclipse, 2006). Other developer tools included fabFORCE DBDesigner 4 (fabFORCE DBDesigner, 2006) which was used for data modelling, as it was the best open source data modelling package found. We did discover some problems with this application, like a bug in the user interface causing the Table Editor to freeze up when editing table specifications by using the mouse to change focus, but these problems could be worked around.<sup>2</sup> Both Putty (for secure shell access) (Putty, 2006) and WinSCP (for file transfer) (WinSCP, 2006) were indispensable tools for managing the Red Hat Enterprise Linux servers (Red Hat, 2006) used from Windows PCs.

The project also utilised a list of important software running from servers. Subversion (Subversion, 2006) was used as the version control system for source code and documents. This was mainly because it was considered superior to the other common open source choice of CVS (CVS, 2006) by the author of this thesis. One convincing argument being that it keeps track of a whole tree of files, and not just the separate files. This is a major advantage, especially when refactoring the code. For the web server, Apache (Apache, 2006) was used in combination with Apache Tomcat for Java (Apache Tomcat, 2006) and Apache Axis for Web Services (Apache Axis, 2006). MySQL (MySQL, 2006) was chosen for the database server in the system, as it provided the required features. Moreover it is currently a very common choice for web based applications. phpMyAdmin (phpMyAdmin, 2006) was used for database administration; this is a web application that handles the standard

---

<sup>2</sup> The author of that application is now working on a new data modelling package for MySQL AB (the company behind MySQL), so in the future that will hopefully be a better solution (MySQL Workbench, 2006).



administrative tasks of editing and viewing MySQL databases and testing out SQL queries. This was all run on GNU/Linux servers.

The choice of development tools in this project showed that free software tools in most cases were more than adequate to handle the requirements. One exception being data modelling packages, these were not found to be of as high quality as commercial offerings.

## **4.6 Task allocation**

Within the eQuake project, the development team consisted of Lalitha Jonnavithula (Masters student), Yuejun Zhang (PhD student), Jingyu Yang (PhD student), Jianbo Cui (part time programmer) and Øyvind Smestad (author of this thesis, Masters student and part time programmer). The tasks were split as follows:

Lalitha – user interface design

Yuejun – student and tutor proxy agents and notification module

Jingyu – query monitoring agent

Jianbo – target selection agent and study group related functionality and

Øyvind – data layer, dispatcher, proxy agent, forum display (implementation of the user interface, except study group functions) and LMS plug-ins.

Øyvind was also responsible for coordinating the development and designing/specifying the first version of the interfaces for each of the components to implement.

A larger group more or less acted as customers who helped specify the requirements, suggest possible solutions and test developed prototypes. The graphical design was largely conducted by Online Learning Systems, who was an external partner.

## **4.7 Team communication tools**

Face to face meetings were used as the primary communication tool within the development team. The project group as a whole used some face to face plenary meetings, but the majority of the discussions took place in an online discussion forum. For communicating ideas, several diagrams describing the

proposed system were developed. This as a means for all participants to get a clearer picture of how the system would be developed, once most of the requirements had been identified and written up in requirements specifications. These included both system component diagrams (as shown in figure 5) and entity relationship (data structure) diagrams (figure 9).

The next chapter will present the architecture of the system. It starts with the system architecture including a thorough explanation of how each system component works, then follows with a description of the data architecture chosen and finishes with an explanation of possible deployment architectures.

## 5 eQuake architecture

The architecture of the system was mainly developed with the goal of supporting interaction between students in different courses and institutions. This by setting up shared forums as shown in figure 4. The figure shows an example of an eQuake learning forum being shared by one course at Massey University and one course at Eastern Institute of Technology (EIT), as opposed to the traditional way of having separate discussion forums within each course. The other requirement that in the end had a significant influence on the selected architecture was the demand for each institution to physically be in control of data contributed by their staff and students.

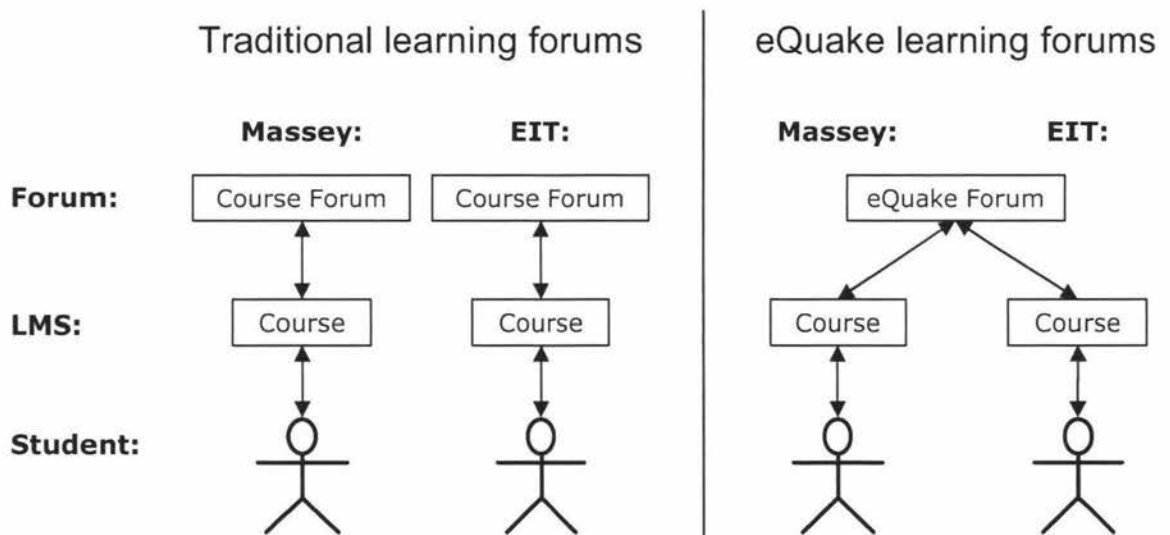


Figure 4: Use of traditional forums vs. eQuake forums

In this chapter the design of the architecture is discussed from three different levels: system architecture, data architecture and deployment architecture. It is important to note that the design decisions within these are heavily entwined, and should be seen in combination.

### 5.1 System architecture

How the system is broken up into components is shown in figure 5. The system consists of three layers, Presentation Layer, Application Layer and Data Layer. Boxes signify components while cylinders are databases and lines show which components interact with each other. Lines marked with *WS* are where Web Services are used to communicate between components. Greyed

out parts are not developed in the eQuake project but are included to show how the system integrates with an existing LMS. Dashed parts signify components not yet developed, but which are suggested as future expansions. All system components are developed in Java, with the exception of LMS Plug-Ins and the Forum Display component. Following is a description of the functionality provided by each component.

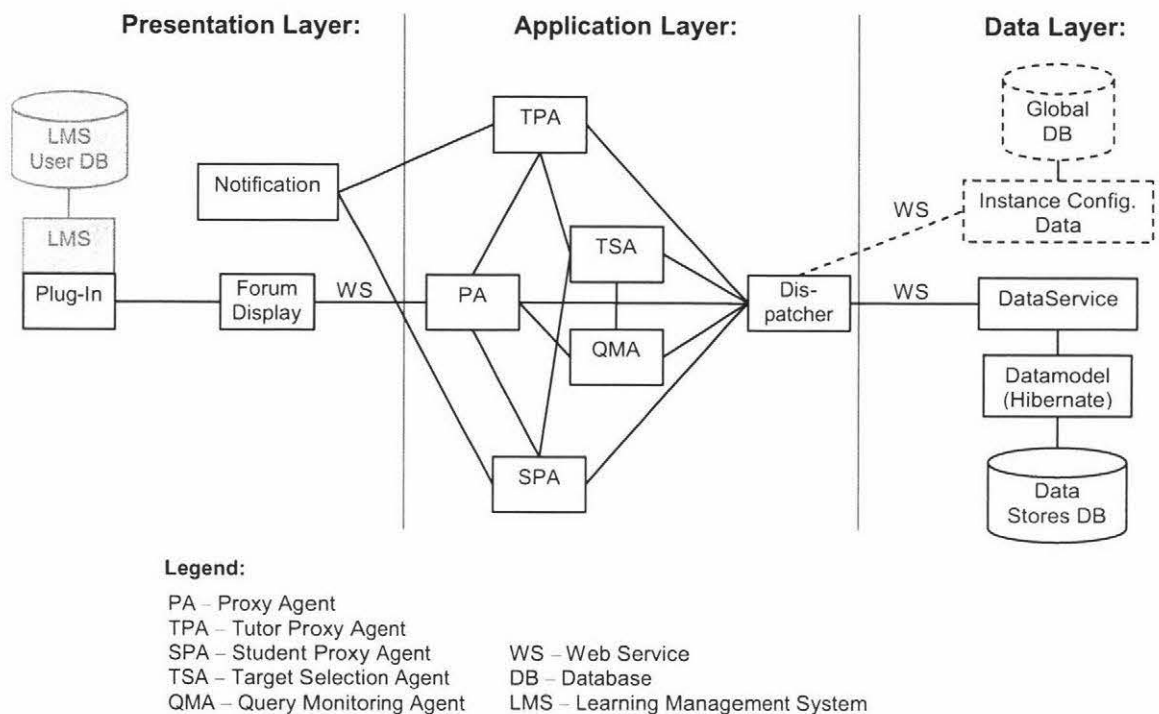


Figure 5: System diagram

### 5.1.1 LMS Plug-Ins

Two plug-ins have been developed to integrate the eQuake forum system with existing learning management systems, one plug-in for the open source Moodle system and one plug-in for the proprietary WebCT system. This was done to show that the system is technically capable of integrating with both open and closed environments.

The main task of these plug-ins is to log users in to the eQuake system automatically when they are already logged in to their respective LMS. This is done to save the users from having to log in twice, which would be both irritating and potentially confusing. The plug-ins also enable the eQuake system to keep the user data in synchronisation with the hosting LMS.

The version of WebCT currently being used at Massey University is Campus Edition version 4.1. This system is developed in the Perl language (later versions are completely rewritten in Java). The official documentation for the Application Programming Interface (API) this version exposes to developers for creating extensions (plug-ins) to the system is only available from WebCT for licensed users. A document describing this API was though found by searching the Internet. Developer or administrator access to a WebCT server to be able to develop and test a server side plug-in based on this was however not available. It was therefore decided to try implementing a simpler way of forwarding the required information to the eQuake system without the need of server side scripting.

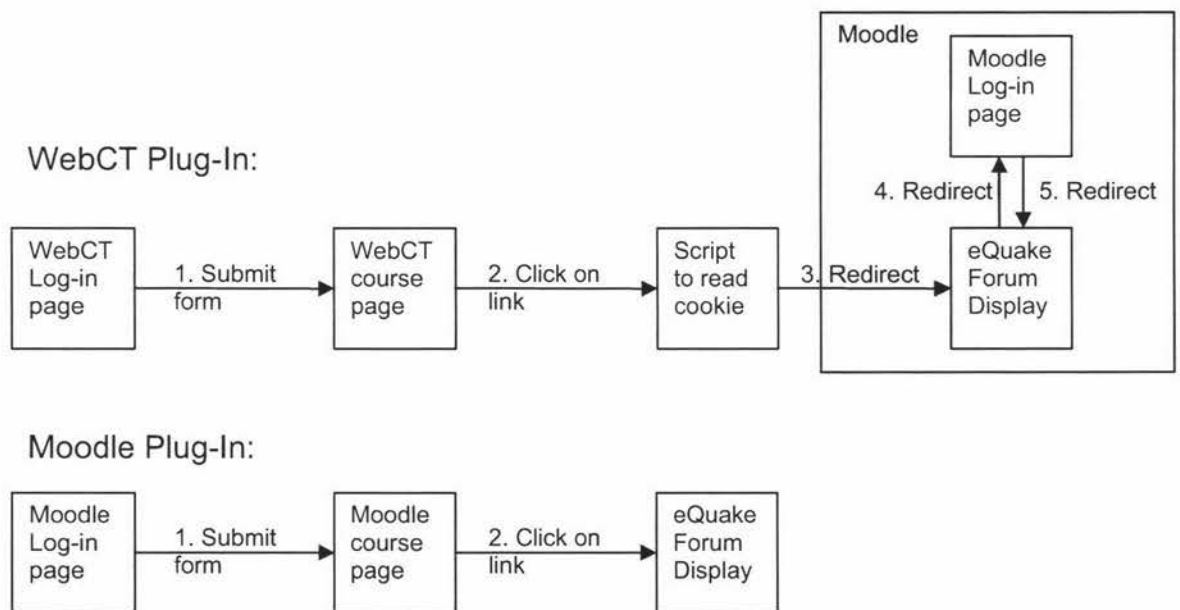


Figure 6: Control flow when logging in to an eQuake forum

A plug-in based only on client side scripting was successfully implemented and the solution can be seen as the WebCT Plug-In part of figure 6. This works as follows: when users are logged in to their WebCT accounts and are accessing a course that uses eQuake forums, they have a link, for instance named *Discussion Forum*. Clicking on this calls a webpage containing JavaScript. Running on the users' computers this script then reads the Cookie set in their web browsers by WebCT. There it finds the *userId* of the users and forwards that along with the ID of the eQuake forum to the Forum Display

part of the eQuake system by using a redirect. This component then logs the users in by using more redirects and in turn displays the requested forum. This is not a very secure solution, in that by knowing how the system works and guessing users' *userIds* it would be possible to log in as them. But we consider this approach good enough to prove that our prototype is working.

To install a connection to an eQuake forum in a WebCT course, one HTML file containing the URL of the Moodle server, the ID of the forum to access and a small script written in JavaScript has to be uploaded to the WebCT course. This can then be exposed to the students as a link or button named *Discussion Forum* or something similar.

It should be noted that when using eQuake with WebCT a Moodle server is still required because the Forum Display component requires some functionality provided by Moodle. This is indicated by the grey Moodle box in the WebCT part of figure 6.

For integrating directly with Moodle on the other hand, eQuake Forum Display is installed as a module inside Moodle and works as a fully integrated part of the Moodle system. Accessing an eQuake forum then happens as in the Moodle Plug-In part of figure 6. Since Moodle here is used directly to track users, no extra login is required to be done by the plug-in.

In Moodle teachers can create links to eQuake forums by adding an instance of the eQuake forum module which is available in their *Add an activity* menu.

Both Plug-Ins allow eQuake to be integrated with the respective learning management systems. Though as Moodle was the primary LMS selected to integrate with for the project, that plug-in has a more fully developed administrative interface and is thus easier to set up.

### **5.1.2 Notification**

The Notification module is responsible for pushing notifications out to the users of the system. Currently all notifications are sent by e-mail. This could

possibly be extended to other mediums such as text messages to mobile phones or personalised RSS feeds. The medium could then be selected based on user preference and estimated importance of the notification.

### **5.1.3 Forum Display**

The Forum Display component is the implementation of the user interface of the system. In other words, it generates the HTML that is sent to a user's web-browser. It is written in PHP and was based off the existing forum module in Moodle. That was first changed to interact with the eQuake Proxy Agent through a Web Service connection instead of accessing data in the Moodle database. Later the message types were added, the layout of discussions was changed and all other parts of the user interface were updated to expose the added functionality of eQuake. The final version is very different from the original Moodle forums, but it was a good choice to start with that code, as a user interface was then available to test the system with right from the early stages of development.

### **5.1.4 Proxy Agent**

The public methods of the Proxy Agent are exposed as Web Services and this is the point in the Application Layer that receives and responds to all user requests through the Forum Display component. Upon incoming requests the Proxy Agent checks what the user wants and calls the correct component of the system. When it receives a reply, it forwards this to the Forum Display that in turn displays the result to the user.

### **5.1.5 Tutor Proxy Agent**

The Tutor Proxy Agent implements the functionality that is specific for tutor or teacher users. This includes notifying tutors of new groups of similar questions, so a new Frequently Asked Question (FAQ) can be added to the system. The component also stores the new FAQs in the system when entered by a tutor and it notifies the tutor if any students show signs of problems with understanding.

### **5.1.6 Student Proxy Agent**

The Student Proxy Agent logs all student activities. It notifies students of new messages that they are subscribed to and can also notify them of



recommended new answers and study groups that are found to be possibly interesting for them based on their student profile. It also monitors the students and can send them notifications if their participation is low or if other signs of possible problems with their learning are discovered.

#### **5.1.7 Target Selection Agent**

The Target Selection Agent is used to search the tutor profiles of tutors in order to ask them to create new FAQ entries on a given subject based on how their specified proficiency areas relate to the identified group of similar questions. It also does the actual searching of the student profiles to find students who might want to join a specific study group or who might want to read a new message, based on what the system knows about the students' interests.

#### **5.1.8 Query Monitoring Agent**

The main task of the Query Monitoring Agent is to find possible existing answers in the system when a new question is asked, as seen in figure 7. This diagram shows the data flow between the student and the system when a question is asked, excluding all intermediary components. When the Query Monitoring Agent receives a question it first searches for similarities with the FAQs stored in the system, then with student contributed messages and finally with other unanswered (student contributed) questions. To make this more efficient, it extracts keywords from all new messages and stores them with the message when the messages are added to the system. It also monitors the similarity between posted messages and can suggest notifying a tutor when it finds that a group of messages have been made in a certain area, indicating that there might be a need for an FAQ within that area. The current comparisons are quite simplistic in that all words, except a pre defined list of non-content carrying words, are considered keywords. All words are currently given the same value of importance. Improvements to this could for instance be stemming the keywords (using only the root form of the words), or using techniques to weight keywords based on how commonly they appear in the forum like tf-idf (term frequency – inverse document frequency) (Salton & Buckley, 1988).

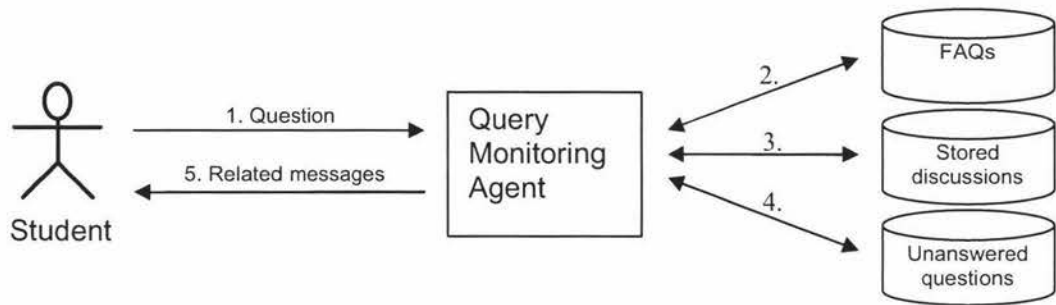


Figure 7: Query Monitoring Agent interacting with student

### 5.1.9 Dispatcher

The dispatcher receives all calls by the Application Layer to retrieve or manipulate data. It then uses Web Services to call the Data Service of this instance of the system, and of any other system instances that are part of the same forum. After gathering and sorting the results they are returned to the calling agent as one result set.

### 5.1.10 Data Service

The Data Service exposes the Data Layer as a Web Service that can be accessed both by the local instance of the system, and any remote systems that take part in the same forum. This component uses a programming method called reflection to translate the incoming requests to the objects that can be used with Hibernate as specified in the Datamodel package. It then uses Hibernate to access the actual database.

### 5.1.11 Datamodel

The Datamodel package contains the Java persistency classes and XML descriptions used by Hibernate to map database tables to Java classes.

### 5.1.12 Instance Configuration Data

The Instance Configuration Data is a proposed component that would be used for administrating the forums and for specifying which courses participate in each forum. This would for instance enable a course to join an existing forum and doing so without editing the configuration files of all system instances already participating in that forum. This component should most likely be exposed in a separate user interface for administrators only.

Splitting the system up in components with clear responsibilities and defined interfaces like this, helped partition the work among the development team.

## 5.2 Data architecture

The structure of the information in an eQuake forum is seen as the white boxes in figure 8. This structure is quite similar to what is commonly used in other web forums. A *forum* defines a wide subject area that can contain several *topics* which are more narrowly defined areas that again each can contain many *messages* or *posts*. Students are allowed to add messages, while topics are set up by the teachers or tutors. A range of forums would ideally be set up centrally using an agreed upon ontology of subjects. This is to encourage more institutions to participate in the forums already defined and not create their own.

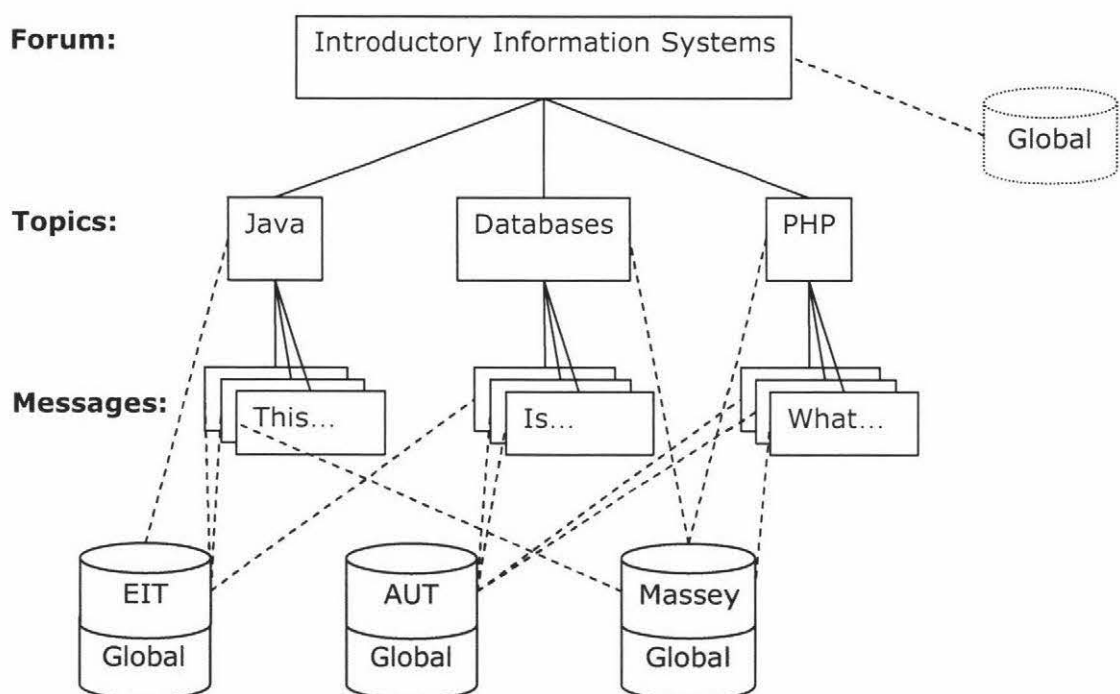


Figure 8: Information structure and storage structure

Messages are stored in threaded form. This is not depicted in the figure as these links between the messages within a topic are not shown. What this means for the structure is that each message, except for *discussion starters*, are linked to the message they are replying to. This in turn is used by the system to know which messages belong together and form a discussion or *thread*.

The dotted lines and grey cylinders in figure 8 show an example of one possible distribution of how data can be stored. What is special with the eQuake system in this regard is that all data is not stored in one central location. The data is only stored in the instance of the system of the user who entered it. For example, a message by a Massey University student will be stored in the Massey database, while a reply to this message by an EIT student will be stored in the EIT database. To generate the view of a forum the system therefore needs to access all connected servers to retrieve all the related information. The actual forum definitions are here considered global in that they are replicated in all the databases. There is, in other words, currently no physically separate global database.

The information structure of the eQuake forums is quite conventional for web forums, but how it is stored with a distributed data structure is not common. This is implemented to allow each institution to be physically in control of their data.

### 5.2.1 Data model

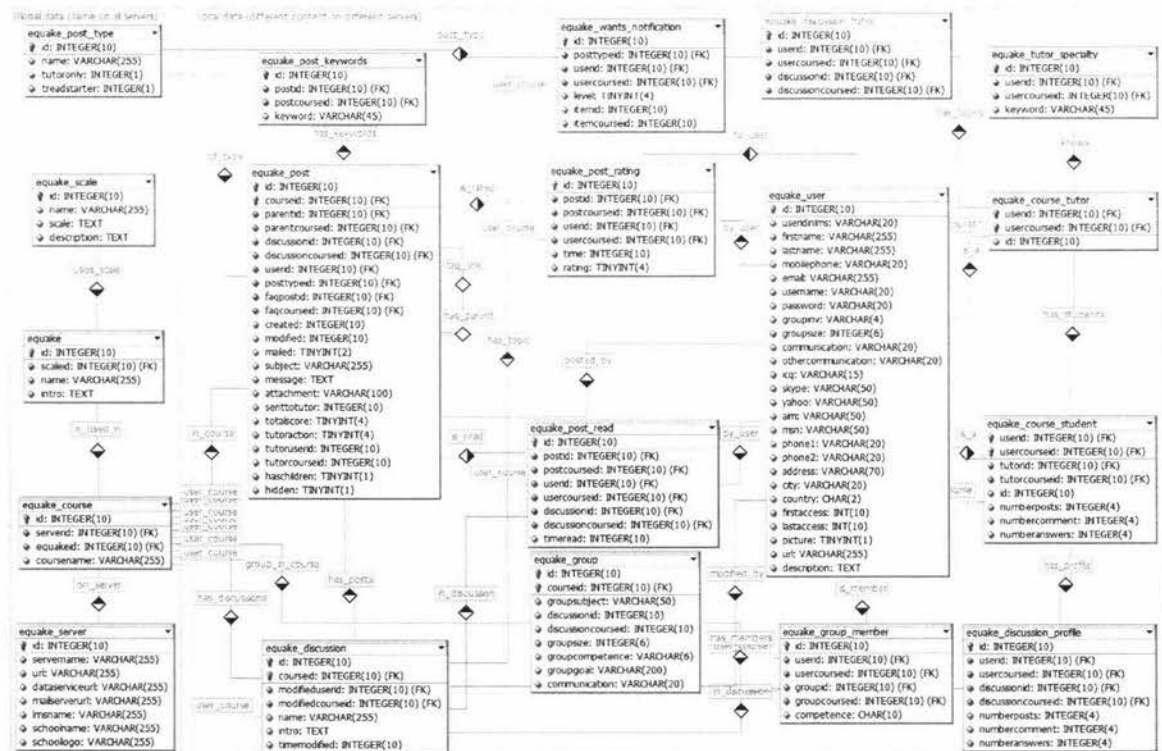


Figure 9: Entity relationship diagram

The actual entity relationship (ER) model for the databases used is shown in figure 9. It is separated in two parts; one is the *Global data* on the left which is replicated in all servers running the system. This part also has the option of being kept in a separate global database to ease the set up of new servers and to synchronise updates to the data in these tables. The second part is the *Local data* to the right that contains all information about users, messages, topics and groups defined in one instance of the system. This part should only be editable by users from within this instance of the system. A full description of each of the entities is available in Appendix A.

Keeping the local data protected from changes by users of different instances of the system works in most cases, but some exceptions have had to be made. One example of a task that would not be supported in this design is for a teacher in one course to perform actions on a message posted by a student in another course (on a different server), this as tutor actions are stored with the message in question. This becomes a problem as the eQuake system is intended to provide the teachers with various options for splitting the workload of monitoring the forum between them. Teachers can choose to look after only their *own* students and in that case, all work is done without problems on the local server. But if they instead split the topics in the forum between them and each moderate their respective topics, then the system must enable them to perform actions on the messages of students on other servers. So, here it was necessary to implement an exception to the rule that data in the *local* part of a database was not editable from outside. This approach works well in that it does not have any data consistency problems, since all data is still stored in only one location. To the teacher, it makes no difference if they are performing tasks on a locally stored message or a remotely stored one.

This exception could be avoided by having a separate table with message metadata and keeping that in the global data section. But this would make many database queries much more complex and would completely change how the global data section is used. This solution would improve the data

integrity in that local data would be secure from external changes, but if was considered as adding too much complexity and also likely to reduce performance so it was not selected.

A third option would be to add a separate table in the local data section for message metadata, causing it to then be stored only in the tutor's instance of the system. This would cause the system to have to check all server instances connected to a forum for tutor *actions* on a message before displaying it. This would require another round of database queries and is therefore likely to reduce performance of the system. This model would also allow a message to be tagged with several tutor actions at once, something the current system is not intended for. This can be seen either as positive or negative, but would in any case require more complex logic for handling which actions can be combined and which take precedence. How tutor actions are displayed in the user interface would also require change. The added complexity and possible reduced performance caused this option not to be selected.

### **5.2.2 Data access layer**

The fact that the data is distributed is hidden from the core system. This is achieved by all data accessing calls going through one component, called the Dispatcher, in the Application Layer. This component handles requests to retrieve data and to change data in two different ways. Requests to store, update or delete data are sent to the local instance of the Data Layer by using its Web Service (that sits on top of the local database). As discussed previously some exceptions to this exist that allow modifying specific fields of some tables on remote system instances by some users (for instance tutor actions on a message). Requests for retrieving data, on the other hand, are forwarded to the Web Service of all Data Layer instances that this system is connected to (all institutions participating in this forum) and the results received are then combined before being returned to the calling component.

## **5.3 Deployment architecture**

Two approaches have been identified and evaluated for the deployment of the system. The first architecture that was explored consisted of using a central



database to store all data in the system. The Application Layer (eQuake System) would run on the central server and the Presentation Layer (eQuake GUI) on each of the collaborating institution's servers, communicating with the central server using Web Services, as seen in figure 10.

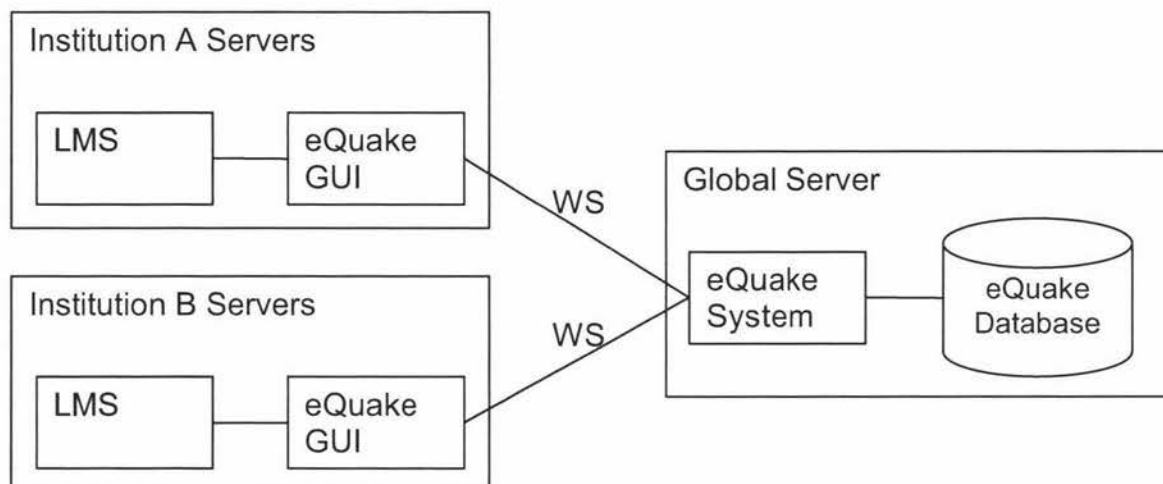


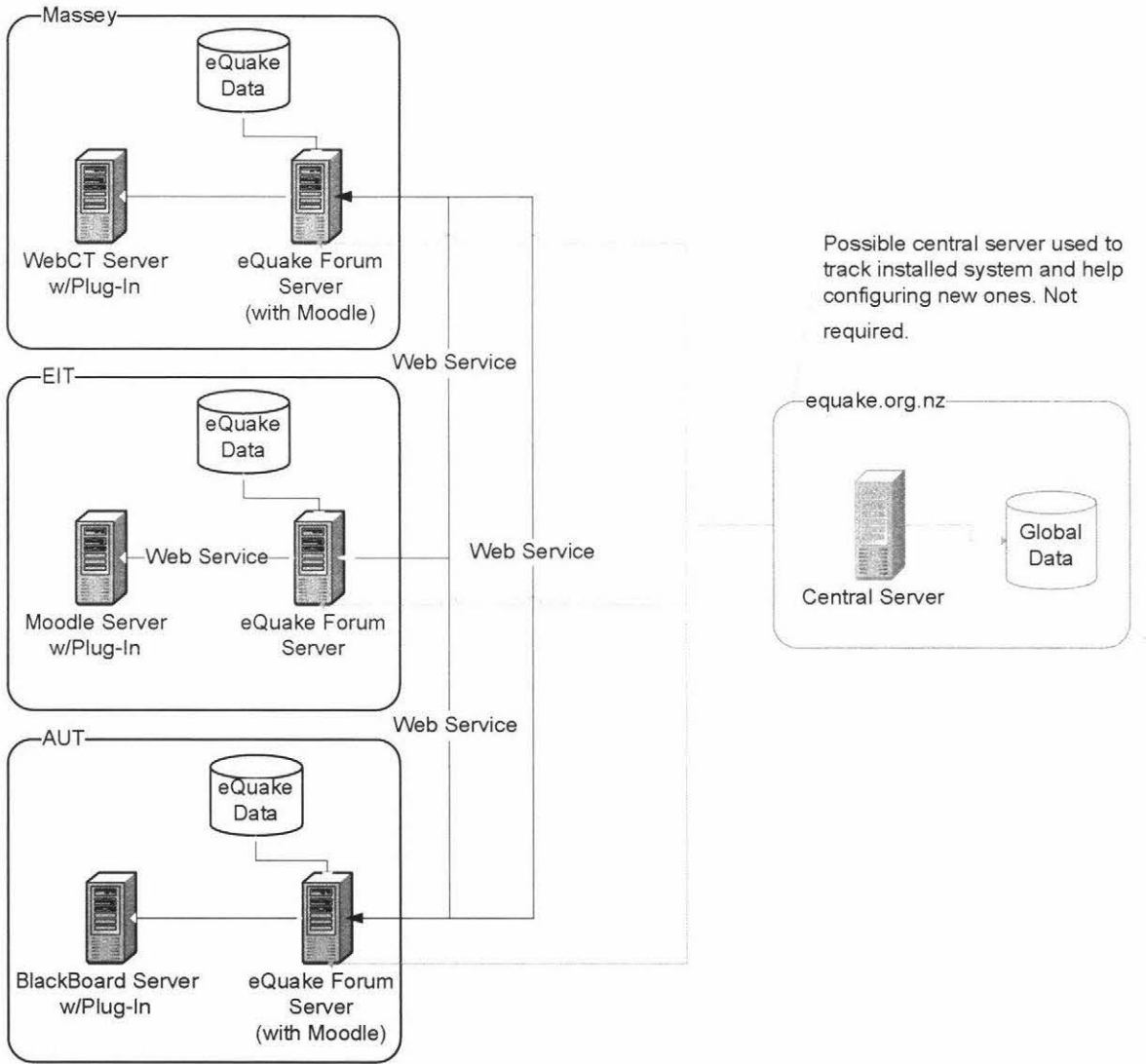
Figure 10: Deployment with only a central database

This service oriented approach where everything except for the LMS Plug-In and Forum Display (GUI) components can be run centrally has the advantages that all data is stored in one place, and most functionality is also located there. Maintenance of the system would therefore be easier, and scaling the system could be done by adding resources at one point. The negative sides to this architecture is that it might require large central resources to scale well and institutions might not be willing to store “their” data in a central location, especially if that central server is hosted at another institution.

The second architecture, on the other hand, places everything within each institution. An example of a possible deployment of the system in three different institutions is shown in figure 11. The data structure used here is a homogeneous distributed database (Bell & Grimson, 1992) as the structure of the data stored at each physical site is the same, but the content of the data is different. The LMS Plug-In or the eQuake GUI (Forum Display) communicates with a local eQuake server that has a local database which stores messages by local users and any tracking data and preferences for these



users. This server also communicates with all other servers that contribute to the same forum using Web Services, forming a peer to peer network between the servers. This is used when retrieving data, as requests are then sent to all connected instances. A possible central server is also proposed to keep track of the list of forums and the institutions that are using each forum. This would especially be used for connecting new institutions to the network in a simple way, though the system also works without this server. For performance and reliability, these *global data structures* are anyway duplicated in each running database of the system, as they rarely change but are often queried.



first approach might be too radical for getting real life support. The hardware requirements for this system are also more evenly distributed among the collaborating institutions, as no central server is required. The system is also more likely to still be usable inside an institution even if their local network is temporarily disconnected from the Internet. When looking at scalability, it can also be argued that this solution is very well fitted for large amounts of data. Since the database is not restricted to be able to fit on one physical machine, this approach is somewhat similar to what Google uses to distribute data and processing among several servers (Pike, Dorward, Griesemer & Quinlan, 2005). However, the forums will have to grow quite large before this becomes a selling point.

The distributed architecture selected was the only option that fulfilled all of the requirements defined for the system.

### 5.3.1 Flexibility of possible deployment

The developed eQuake system is very flexible in how it can be installed on different servers.

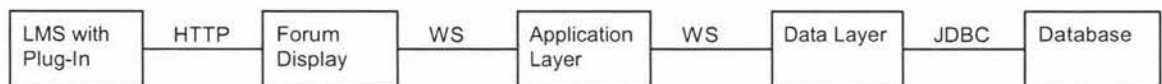


Figure 12: Layers of the eQuake system

As seen in figure 12, the eQuake system can be seen as a five layer architecture, where each layer represented by a box can be installed on a separate server if required. The lines between the boxes signify the communication channel used between the layers. In the case of WebCT as LMS, an HTTP GET request is sent from the LMS to launch the Forum Display. If the LMS is Moodle, the Forum Display is integrated directly as a module inside the LMS, so the communication is based on PHP function calls. In this case, these two components have to be on the same server. Between the Forum Display and the Application Layer, there is a Web Service connection and this is also how the Application Layer communicates with the Data Layer. The Application Layer communicates not only with the Data Layer within the instance of the system where it resides, but also with the

Data Layers of any other instances of the system that participate in the same forum. The Data Layer then uses a JDBC connection to communicate with the actual Database.

This gives a high degree of freedom in the number of possible ways of configuring and installing the system on servers. This is good as the system can be tailored to take advantage of available server resources. If for instance, one part of the system shows CPU bound performance problems, it can be moved to a separate server and given more hardware resources. On the other hand, if performance is low because of slow input/output (I/O) performance, components can be moved together on the same server to reduce network delays.

A successful architecture was designed for the eQuake system allowing the specified functionality to be developed. In the next chapter we will look at some of the specific issues faced when implementing this system and the design decisions arising.

## **6 Discussion of implementation issues**

This chapter will look at some of the issues faced while implementing the system. Some central sides of the user interface design are also discussed where these interlink directly with other decisions. The user interface is though not described in full detail as that was outside the scope of research for this author (a forthcoming master's thesis by Lalitha Jonnavithula will fully address this side of the system).

### **6.1 Discursive learning environment or intelligent help system**

The goals of providing an environment to support discursive learning and providing an intelligent help system can in some cases be considered as opposing forces. This can for instance be the case when considering how many old messages should be available in the system. For being an intelligent help system, having available as much information as possible gives a higher likelihood of finding the correct information. But to encourage students to discuss, it might be better to hide what previous students of a course have contributed, as it might be hard to come up with new points if everything has already been covered in previous messages. This could possibly cause students of later iterations of a course to not contribute any more.

eQuake » Massey » IS Forum » HTML (Building web pages) » How to show pop-up windows?

Start new discussion:

Subscription: [Subscribe me to this topic](#)

[Study groups in topic](#)

Foundation by Sylvia Chen - Thu, 18 May 2006, 04:06 AM

How to show image by Tsai Chun-I - Thu, 18 May 2006, 06:26 PM

Re: How to show image by David Latham - Wed, 31 May 2006, 10:38 AM

Create new study group

**How to show pop-up windows?**

by Huang Meng-Sheng - Friday, 19 May 2006, 04:48 AM

Can HTML show pop-up windows, and what is the function(or tag)??

Can somebody give me an example? 😊

Tutor action:

Post is seen Tutor: Michael Verhaart

Re: How to show pop-up windows? by M V - Fri, 26 May 2006, 09:44 AM

**Legend**

**Post types:**

- Answer
- Comment
- FAQ Answer
- FAQ Question
- Information gathering question
- Solution seeking question
- Suggestion / Idea

eQuake » Massey » IS Forum » HTML (Building web pages) » How to show pop-up windows?

You are logged in as Oyvind Smestad (Logout)

Massey

Figure 13: eQuake user interface

Another question that might be affected by this is how many messages should be displayed in full within one topic or discussion. For eQuake, as seen in figure 13, we chose to display one message in full with the headers of the related messages above and under that message. This makes the display less crowded, while showing a message in its context, and also makes it possible to track which messages a user has read (or at least viewed). On the other hand, it might make it harder for the users to get an overview of a whole discussion, since they actively have to open each message to read it, and might need to go back and forth between viewing different messages to compare them.

When developing eQuake we tried to balance the features of the system so that it can be used both to support a more *questions and answers* focused help environment and to be an environment that can encourage deeper discussions. Some of these issues are more related to how a forum is run than how it is implemented though. But as eQuake forums are intended to be used by multiple courses at once, it might not be possible to *clean out* a forum between iterations of a course as can be done when using a traditional, in

course, discussion forum. To see if this becomes a problem would require longer term evaluation of the system.

## 6.2 Integrating message types in the user interface

How the users select message types had to be integrated as seamlessly as possible in the user interface. This was done so that users familiar with traditional discussion forums did not find this additional requirement to be a nuisance. The approach taken is similar to what is implemented in ForumPlus (Calvani et al., 2005) with a dropdown menu to select the type of message, instead of a link or a button to *Reply*. This dropdown menu, as seen in figure 14, was then placed where users would expect to find means for starting a new discussion or replying to an existing message. After making a selection in the dropdown menu they are taken straight to the *enter message* page; there is no button that needs to be pressed after making the selection (this is achieved using JavaScript). The solution selected has proven successful, and users have adopted it willingly.

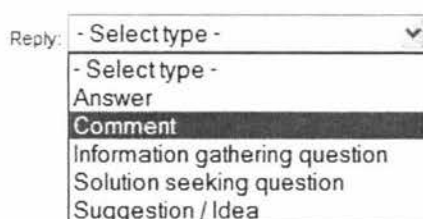


Figure 14: User interface for selecting message type

## 6.3 Navigation of large message archives

An issue with having a large forum with many topics and many messages is that messages are frequently posted in the wrong topics. This can easily happen, as users might not know under which topic to post a certain message. This is a problem as it makes it hard to find messages within a certain area by browsing the topics; in other words the topics might lose their value as navigational aids. It might also take longer to get an answer to a posted question, as the question might have to be moved to another topic by a moderator before someone able to answer sees it.

To improve on the *navigating to find information* part of this problem eQuake encourages using searching (by asking new questions), instead of

browsing all topics, for finding needed information. The second part of the problem, that questions might stay unanswered, is improved upon by the fact that each message is tracked by the system. The system knows which messages are unanswered questions and the *age* of each unanswered question (how long it has been unanswered in the system). This is used to generate lists of the oldest unanswered questions in the system. These lists are then displayed in the user interface to encourage the questions to be answered. This could also be extended further to include escalation after a given amount of time, in that questions not answered within, for instance 48 hours, are forwarded directly to the teacher. Intermediate levels that ask for input from students identified by the system as possibly able to answer the question could also be a possible extension.

These are important problems for a system like eQuake that strives to enable inter institutional discussion forums that inherently will contain large numbers of messages. The solutions currently implemented will hopefully help solve these problems, but validating this would require larger scale evaluation than presented here to accumulate the number of messages needed before this normally becomes visible problems.

The next chapter presents the formal evaluation of the system and how continuous evaluation was performed throughout the development cycle.



## 7 Evaluation

This chapter will describe the continuous evaluation of the system that has been carried out throughout the project and the formal evaluation that was performed at the end of the development cycle. The main formal evaluation was primarily with student users, but a minor evaluation with only teachers was also performed.

### 7.1 Continuous evaluation

One of the advantages of choosing to base the user interface on an existing forum solution (Moodle forum), was the ability to quickly have a prototype version of the system up and running. This was helpful by allowing the larger project team to test the system and thereby give continuous feedback throughout the project duration. The prototype was also a good supplement to diagrams and documents to communicate ideas and show the status of the system during development. Feedback on the system was received both in a dedicated web discussion forum and in project meetings with all the project stakeholders. The feedback included both requests for additional features, changes to existing features and reports of bugs. This way, changes could be made to the system as and when concerns were raised and new improved versions were then made available to be tested. A prototyping approach with continuous feedback was seen as an integral part of the development strategy, allowing rapid improvements to take place.

### 7.2 How the system was set up for formal user evaluation

For the formal user evaluation of the system with students participating, a total of three *courses* were set up running in two instances of the system, as depicted in figure 15. Both of these instances used Moodle (version 1.5.2) as the LMS to access the eQuake forum system through. One course in each instance of the system was set up to evaluate sharing one forum (the Information Systems forum) between two different courses. The third course was used with its own separate forum. The two system instances were both installed on the same web-server, but with separate databases and using separate LMS installations. This was done to test interacting between different servers, as that was one of the main design requirements for the

architecture of the system. In a real deployment these instances would not be located on the same server, but deployed in the institutions using them directly. For the evaluation it was decided to keep all instances in one physical location, as that simplified deployment and gave closer control if any urgent problems with the system would arise.

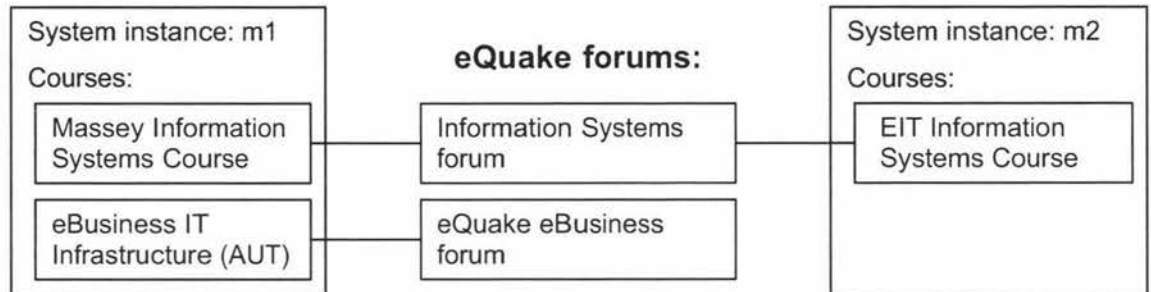


Figure 15: Set up for evaluation

The main forum (Information Systems forum in figure 15) was set up with the following topics:

- Databases
- Dynamic Web Sites
- Electronic commerce
- HTML (Building web pages)
- Multimedia
- PHP & MySQL
- PHP Programming
- Web Markup Languages
- XHTML
- XML (eXtensible Markup Language)
- Feedback about the eQuake discussion forum system

### 7.3 Participants

The system was tested by about 200 users, spread out over the three *courses* that were set up. 113 students, teachers and other interested participants in the *Massey Information Systems Course* and 50 students and teachers in the *EIT Information Systems Course* participated in one forum called *Information Systems forum*. The second forum, named *eQuake eBusiness*

*forum*, was tested by 20 students and teachers in the *eBusiness IT Infrastructure (AUT)* course.

The students were mostly first year (100-level) students, but also some from second and third year. All students were taking Information Systems courses. The other participants mentioned in the *Massey University course* were mostly researchers with an interest in educational technologies.

## **7.4 Methodology**

The methodology used was a survey to evaluate the usability and the users' acceptance of the system. This was done using the Technology Acceptance Model (TAM) (Davis, 1989).

### **7.4.1 Technology Acceptance Model**

Technology Acceptance Model (TAM) is a commonly used model for evaluating users' system acceptance (King & He, 2006; Lee, Cho, Gay, Davidson & Ingraffea, 2003; Legris, Ingham & Collette, 2003; Liu, Liao & Peng, 2005). TAM is used by researchers to predict and explain user acceptance of information technologies. This model has been tested in many empirical research studies and the tools proposed perform statistically reliable results (King & He, 2006).

As described by Davis (1989), TAM is an application of the Theory of Reasoned Action (TRA) (Fishbein & Ajzen, 1975). TAM postulates that behavioural intention is the major determinant of usage behaviour (Davis, Bagozzi & Warshaw, 1989) and that people's computer use thereby can be predicted reasonably well from their intentions to use the new technology. Moreover, TAM suggests that users have a positive attitude towards the technology if they perceive the technology to be useful and easy to use. Therefore perceived usefulness and perceived ease of use are theorised to be fundamental determinants of future system use and of future user behaviour (Davis, 1989). Davis (1989) defined perceived usefulness as "the degree to which a person believes that using a particular system would enhance his or her job performance". Perceived ease of use refers to "the degree to which a person believes that using a particular system would be free of effort". A

system high in perceived usefulness and in perceived ease of use is more likely to be accepted by users.

Figure 16 shows a diagram of TAM: perceived usefulness (U) is a major determinant of people's behavioural intention to use (BI) information systems (IS), whereas perceived ease of use (EOU) is both a significant secondary determinant of people's intentions to use (BI) IS and affecting their perceived usefulness (U). External variables intervene indirectly by influencing U and EOU.

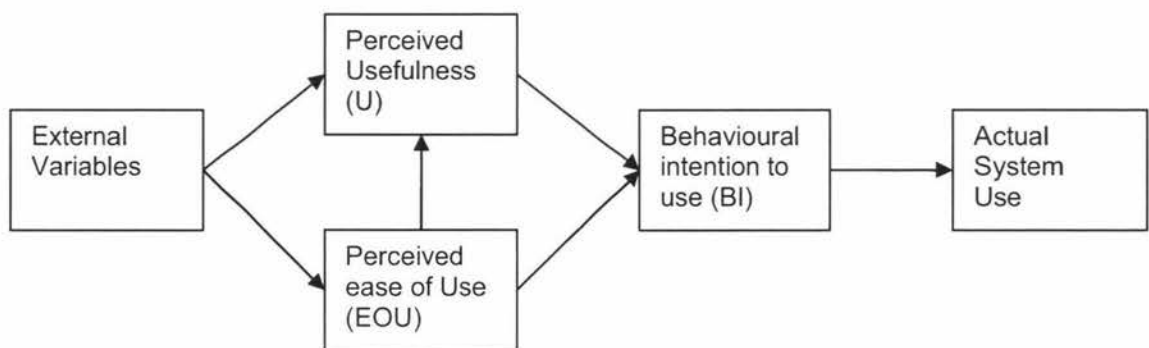


Figure 16: The Technology Acceptance Model (Davis, 1989)

## 7.5 Procedure

The users in our study were all volunteers; they were first requested to use the system and after three weeks they were asked to fill-in an online evaluation form. It was presumed that letting some time pass between introducing them to the system and requesting their feedback would allow them to get a clearer picture of how the system worked as they would have more time to evaluate it. Hopefully more of the users would then also find time to try out the system. For the other researchers participating as users in the *Massey Information Systems Course*, paper forms were used instead of the electronic one. The form itself (Appendix B.1) consisted of 19 questions with a 7 point Likert scale, ranging from *Strongly agree* (+3) to *Strongly disagree* (-3). This was modelled after the original study by Davis (1989), with the addition that the questions regarding Perceived Ease of Use were asked 4 times, regarding different features. These features were considered to be the 4 most important of the system: posting a message, finding an answer, rating an answer and the subscription features. The form also contained one field for textual comments

and a question asking the users to indicate how often they had used the system.

## 7.6 Results

During the period of the evaluation, a total of 121 messages were posted by all users involved. Table 2 shows how these messages were split up in the various message types. As can be seen, all message types have been used. Table 3 lists the numbers of subscriptions to notifications, stored in the system.

Table 2: Messages posted split by message types

Message type	Count
Answer	27
Comment	43
Information gathering question	18
Solution seeking question	9
Suggestion / Idea	12
FAQ Question	6
FAQ Answer	6
<b>Sum</b>	<b>121</b>

Table 3: Subscriptions to notifications by the users

Subscriptions	Count
User subscriptions to a topic	21
User subscriptions to a post	126
<b>Sum</b>	<b>147</b>

Another interesting point to look at in the usage statistics is interactions between users in the two courses that shared one forum. Of the 109 message posted in the Information Systems forum, 27 were replies by a user in one course to a message posted by a user in the other course. Table 4 shows the types of sequences involved in these exchanges, ranked by frequency. Each sequence is identified by the type of message being replied to, followed by ‘->’ and then the type of the reply.

Table 4: Interactions between Massey course and EIT course users

Type of interaction	Count
Information gathering question -> Answer	6
Answer -> Comment	5
Comment -> Comment	4
Comment -> Answer	3
Information gathering question -> Comment	2
Answer -> Answer	1
Answer -> Information gathering question	1
FAQ answer -> Comment	1
FAQ question -> Comment	1
Solution seeking question -> Answer	1
Suggestion/Idea -> Answer	1
Suggestion/Idea -> Suggestion	1
<b>Sum</b>	<b>27</b>

For the survey we received a total of 35 filled in questionnaires. The mean results for each category of questions is summarised in table 5. After the name of each category the number of questions in the questionnaire associated with this category is shown in parentheses. The values could be in the range -3 to 3, so a mean of 0 would signify a neutral stance to the questions in that category. Appendix C.1 contains the values for each question individually.

Table 5: Average grouped values for evaluation questionnaire

Category	Mean	Std. Deviation
Behavioural Intention to Use (2 q.)	1.59	1.280
Perceived Usefulness (4 q.)	1.40	1.202
Ease of Use (Posting) (3 q.)	1.50	1.395
Ease of Use (Finding an answer) (3 q.)	1.22	1.635
Ease of Use (Rating) (3 q.)	1.60	1.370
Ease of Use (Subscription) (3 q.)	1.62	1.382

The users were also asked to indicate how often they used eQuake in the questionnaire, the results of this is reported in Table 6.

Table 6: How often users used the eQuake system

Frequency	Users
More than once a day	1
About once a day	3
4 or 6 times a week	8
2 or 3 time a week	6
About once a week	4
Less than once a week	10
Not at all	3
<b>Sum</b>	<b>35</b>

## 7.7 Discussion of user evaluation

The number of messages posted in the system gives an average of less than one message posted per registered user. This is a bit lower than what we had hoped for, and might be caused by the fact that all students testing this system also had their in-course discussion forums in addition to this system. They might therefore have preferred to raise their questions there, as they had been doing so far in the semester. Another issue causing a slow rate of postings initially might be that the forums started out empty, except for the various topics that were defined, perhaps causing users to be unsure about how the system was intended to be used.

All message types available in the system were used by the users during the evaluation. From this we can therefore assume that users will accept these message types, and that it is not a too large jump from the forums without message types that they were used to.

The usage data also shows evidence of discussions between users in different courses. About a quarter of the messages in the Information Systems forum were replies by a user in one course to a message posted by a user in the other course. We view this as evidence that the inter-institutional forum was successful. The primary types of interactions that took place were: answers in reply to questions and comments, and comments in reply to answers and other comments, as described in Table 4.



For the questionnaire the means for all categories of intention to use, perceived usefulness and ease of use were highly positive. According to TAM, the users' acceptance of the system is then positive and this indicates that the users are likely to use the system further. One observation was that the ease of use for *finding an answer* was lower (1.22, and with a higher std. deviation: 1.635) than for the other points evaluated in the system. A possible reason for this can be that the users had not been given specific tasks to accomplish using the system. Potential topics for discussion within information systems had been set up, but the users were then only told to try the system out. For this reason, the number of messages in the system was limited and the likelihood of getting a correct answer in the results from the Query Monitoring Agent when asking a question was low. We anticipate this would work better in a real use setting where more students would be trying to achieve the same goals and therefore normally would have more similar questions. Nevertheless, lowering the number of false positives (messages that do not answer the question asked) in the results returned by the system when a user asks a question should be a priority.

It should be mentioned that the number of questionnaires we received was low (35) compared to the number of users that registered in the system (200). This could possibly have skewed our evaluation results to be more positive than was perceived by the whole group of testers, if only the most eager users filled in the questionnaire. It is not possible to know how many messages these users posted as the questionnaire was anonymous. But judging by their self reported usage statistics in Table 6, we seem to have gotten a good spread of different usage levels. A possible solution to the low number of replies is distributing the questionnaires closer to or at the same time as when the users are first asked to try the system. Assuming a possible reason for the low number of replies from the users might be because they tried the system in the beginning of the evaluation period, but were unwilling to go back and fill in the questionnaire when asked to at a later date. Further studies might want to look into this or other possibilities for increasing the number of questionnaires returned.

The user evaluation of the system was deemed successful. This as the system performed well, and the feedback received from users were mostly all positive. Improving the results returned when the user asks a question was although identified as an area with room for improvement.

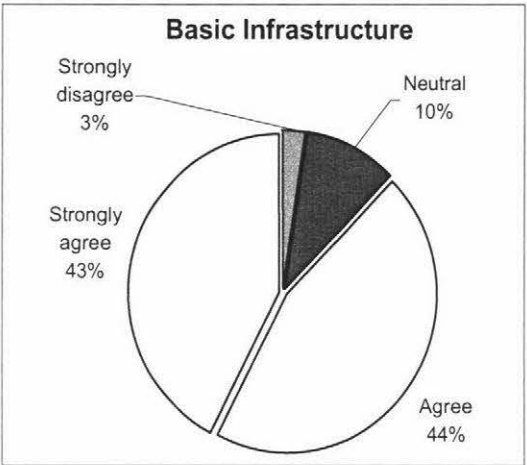
## 7.8 Evaluation by teachers

A separate evaluation was performed with 8 teachers to assess the likelihood of the system getting adopted in real world courses. These teachers were selected from the participants in the main evaluation so the environment for this study is thereby the same. The questionnaire used for this evaluation, which can be found in Appendix B.2, used a 5-point Likert scale which ranged from *Strongly agree* (5) to *Strongly disagree* (1), plus an additional field for not applicable (NA). The questions were split in 7 sections, with sections for the Basic Infrastructure, Learning Community, General Usability and each of the main system components.

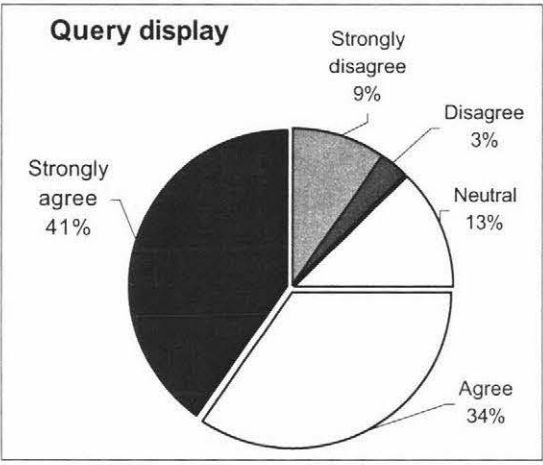
The results of this survey can be seen in figure 17 with pie charts showing the average percentage of respondents in each category for each of the seven main sections in the questionnaire. With the exception of Query Monitoring Agent (d) and Learning Community (f), all sections scored between 75% and 96% for the sum of Strongly Agree and Agree, which should indicate in general good acceptance.

For Learning Community (f), there was a high score for not applicable (NA). This is believed to be because the questions, in that section, were formulated as asking how the system supports the learning communities at each of the individual institutions participating in the evaluation. For instance question 6.1: “The system supports Massey University learning community very well”. We therefore assume that teachers used NA for all institutions except the one they were affiliated with. All other votes were for either Strongly Agree or Agree, so no negative feedback was found here. The teachers are therefore assumed to approve of the support for learning communities in the system.

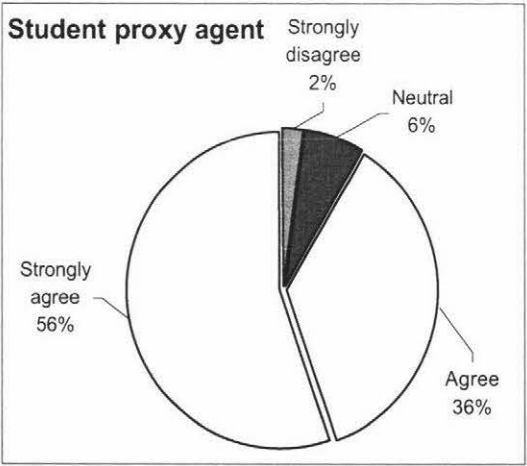
For the Query Monitoring Agent (d) the highest score was for Neutral with 50%. The lowest acceptance for this component is consistent with the results from the student user evaluation and gives more evidence that returning correct results after a question is entered is a key area for future improvement. The highest level of disagreement was found for the Query Display (b). This was the part of the user interface where users could enter questions and view related messages returned by the Query Monitoring Agent. The partially negative result here might indicate that this part of the user interface can be made easier to use.



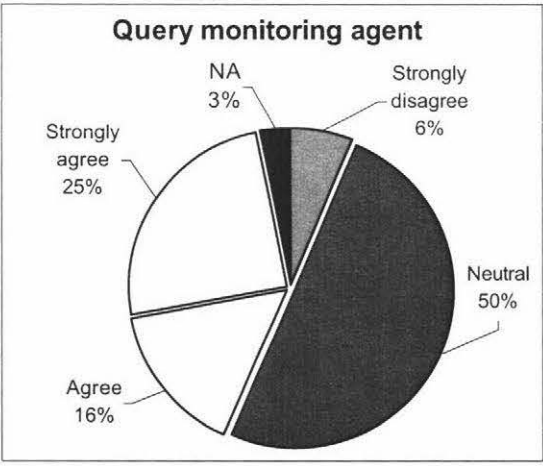
(a)



(b)



(c)



(d)

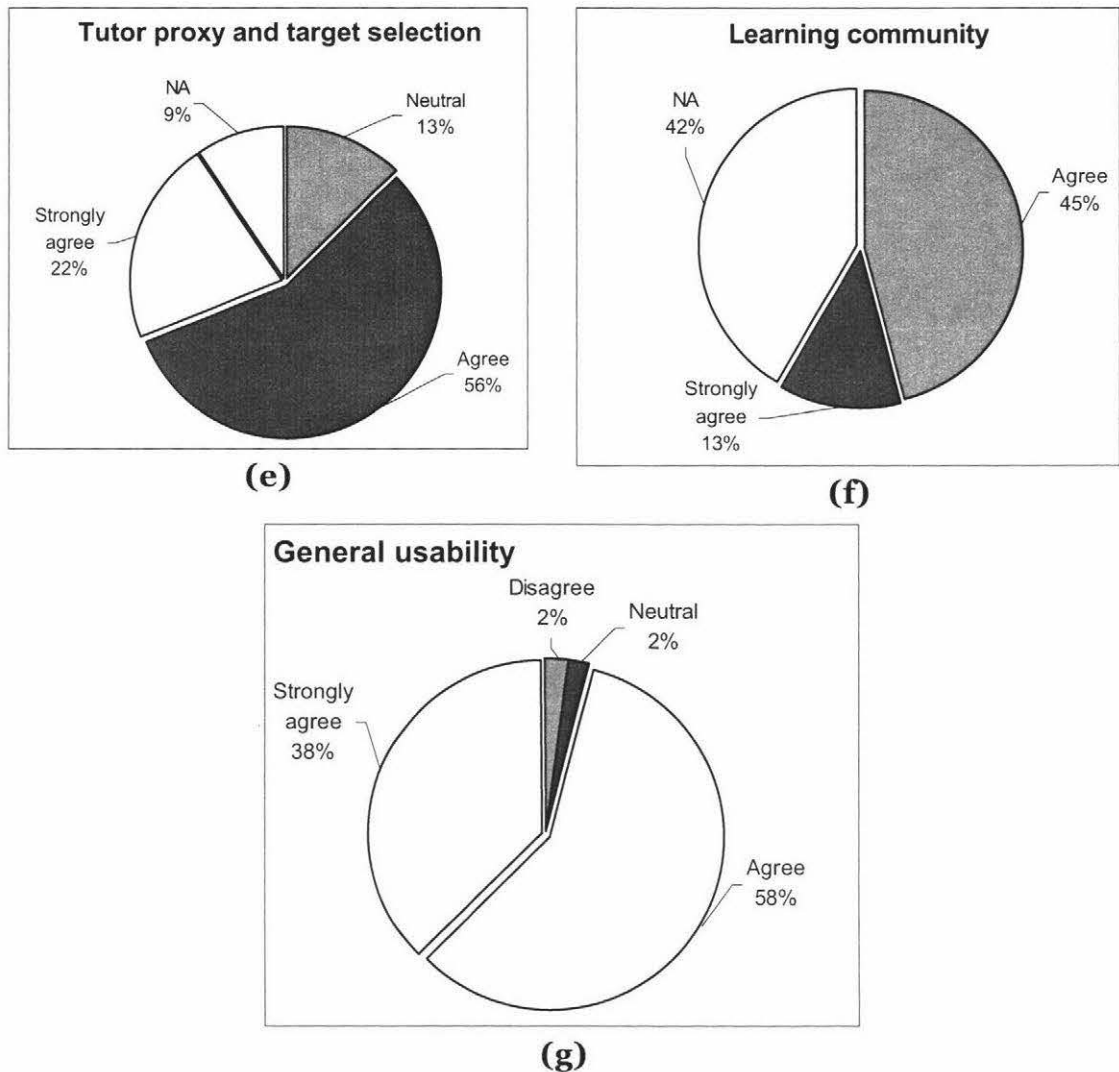


Figure 17: Results of teacher evaluation

To further investigate the possibilities of improving the system, tables 7 and 8 show the satisfaction levels for each of the questions in the sections for the Query Display and the Query Monitoring Agent. Appendix C.2 contains the results for all questions. In table 7 the main problems with the Query Display seem to be related to entering new questions (2.2) and how to then access the different messages returned in the query results (2.3). For the Query Monitoring Agent table 8 shows in questions 4.1 and 4.4 that the searching and results returned by the system can be improved. The low score of question 4.2 is likely because this functionality currently is set up to only activate on identified groups of 5 or more similar questions, and therefore the traffic during the testing period was not high enough for this functionality to be triggered. Question 4.3 similarly received negative results because there is

currently no administrative interface implemented for adjusting *forum level* settings like these. Further development of the system should address these issues identified to ensure continued high user satisfaction.

Table 7: Query Display component questionnaire results

Question	Satisfaction level
2.1 The system allows users to view queries.	100%
2.2 The system allows users to search queries.	62.5%
2.3 The system allows users to retrieve queries.	62.5%
2.4 Overall, the query display system is easy to use and navigate	75%

Table 8: Query Monitoring Agent questionnaire results

Question	Satisfaction level
4.1 The system detects queries pertaining to a question/topic that had been posted.	50%
4.2 The system flags for the frequently repeated queries on the same question/topic and send it with all the related postings to the teacher through the tutor proxy agent.	37.5%
4.3 The system is easy to configure and queries/postings can either be send to all the teachers (who are involved with the subject area) or only to the teacher whose students had asked majority of the questions on the topic.	37.5%
4.4 Overall, the query monitoring agent is easy to use and navigate.	50%

The evaluations of the system both by all users and specifically by teachers were mostly very positive, but highlighted some points for further improvements. The next chapter will conclude this thesis by summarising the findings in relation to the research questions.

## 8 Conclusion

This chapter revisits the research questions raised in the first chapter and summarises the findings in relation to each of these questions. Possible directions for improving the system further are then examined.

The first research question was: *Can a distributed discussion system be developed that enables sharing between multiple tertiary educational institutions?*

We argue that the developed system is the proof that this is possible. Architecture for distributed discussion forums has been successfully designed and implemented, especially since the resulting system was evaluated in a setting showing interaction between students in different institutions using different LMS instances to access the same forum. About a quarter of the messages in the shared Information Systems forum, were from users in one course in reply to messages by users in the other course. This showed that interaction between users in different courses worked satisfactorily.

The second research question was: *Can such a system be integrated with existing Learning Management Systems?*

The developed system works well with the Moodle LMS, which was selected as the primary LMS to integrate with. Integration with the closed source WebCT system has also been demonstrated to work. We therefore conclude that this is possible.

The third research question was: *Would students find the features of the new system useful?*

As the user evaluation was positive and indicated that students would use the system if they had access to it in their further studies, the answer to this question is found to be affirmative in this study.

Research question 3.1 was: *Would students find the system's automatic answer finding behaviour easy to use?*



Based on the feedback received in the evaluation, this feature was viewed as easy to use, but not as easy to use as the other components evaluated. Our assumption as to why this result was received is that the messages returned by the QMA were often not answering the question that was input by the user. This is also supported by some of the written comments received from users. One reason this occurred during the evaluation might have been because the users were only testing out the system and were not using it for real learning tasks. Since the forum had quite a broad range of topics set-up, the number of messages within each area was not that high. Therefore, getting a correct answer returned by the system when asking a new question was not very likely. A real usage scenario where more students are interested in learning the same things and more messages are stored in the system would hopefully increase the likelihood of correct answers returned. But the Query Monitoring Agent should be improved to return less false positives (messages that do not answer the question asked). False positives are common as the current QMA returns the best results it finds, but does not have a minimum score for how low in similarity the returned results can be. Returning false positives because the dataset is initially small, runs the risk of causing the users to lose trust in the system, and stop checking the results returned by the QMA. In that case it might be better to return no results initially.

Research question 3.2 was: *Would students find the system's extended notification features easy to use?*

Judging from the results received in the evaluation, this feature was considered easy to use. Also, table 3 shows that this feature was in active use. However, feedback from users indicates that the formatting of the email notifications should be improved. Currently any HTML code in the message is for instance displayed as text in the notification and it does not include which user posted the new message, or in which topic.

The fourth research question was: *Would teachers find the system useful and be willing to integrate this type of system in their courses?*

The separate evaluation questionnaire filled in by teachers indicates that they find the system in general useful and easy to use. The feature they were the



least satisfied with was the QMA, indicating (as discussed earlier for students) that this is a key area for improvement. In summary, potential uptake among teachers is judged as quite likely based on this evaluation.

The research questions raised in this study were all answered in a mostly positive way, but some areas in need of further improvement in the system were identified. These include the results returned by the QMA when a user enters a question in the system and also how these results are displayed in the user interface.

## 8.1 Improving answer finding search results

This section presents potential ways of improving the quality of the search results returned by the Query Monitoring Agent (QMA) when a question is entered. This is done by proposing various extensions to the search strategy currently employed by the system.

In this explanation the following terms will be used:

- **Textual search algorithm Ranking (TR):** defined as any search algorithm used to compare two texts (the new question and a message stored in the system) and give a score based on their similarity. This can be the current full-text search solution or any other information retrieval algorithm.
- **Message Popularity among All users (MPA):** defines how highly ranked a message is by all users in the system. This can for instance be based on the users' manual rating of messages. Another option is also taking usage data like number of times a message has been viewed into account. This can have a risk in that a message identified by the system as a good answer to a question once will get more *views* each time users' check whether this is correct for their question. Thus causing a message erroneously identified as a good answer once, to also score highly for later similar questions (since previous wrong identifications cause more *views*). While usage data alone would have this problem, combining usage data with manual ratings should solve this.

- **Message Popularity among Class users (MPC):** defines how popular a message is with users in the same class or course as the user currently asking a question (whom the search results will be returned to).
- **Final Ranking score (FR):** output of improved algorithm.

For all the algorithms presented it is assumed that the various scores are scaled to be in the same range (for instance 0.0 to 1.0), and that weights might be added to control how much influence the various parts will have on the final result. This would likely require some experimentation with an implemented system to tune the weights to produce the desired results.

The current approach (in equation 1) only uses textual searching and considers each message stored in the system individually. It uses the message types to filter and group the returned results (FAQs are for instance grouped first). This algorithm does not require any knowledge of the user the results will be returned to, the topic of the messages involved or how popular the messages are.

$$(1) \quad FR = TR$$

Since one eQuake forum can be shared by students in multiple courses it might be desirable to automatically separate course specific messages from topic related messages. The goal of this would be to make messages related to the course of the current user more visible while making messages specific to other courses less visible. One possible way of doing this for the results returned after a question, is presented in equation 2:

$$(2) \quad FR = TR + (MPC - MPA)$$

This causes messages more popular with the class (MPC) than with everyone (MPA) to get an increased ranking score. While messages identified as less popular among the classmates than for all users will get a deduction from their ranking score. Another (most likely undesirable) option is giving bonus points to messages by users in the same course as the current user. These

types of approaches do run a serious risk of constraining topic related discussions within the different courses as well, which is the opposite of what the eQuake system aims to encourage with inter institutional forums. Whether these types of additions would have value is therefore debatable.

Just using the popularity score for all users is another simpler option allowing the users of the system to have more influence on the results returned. This would look something like:

$$(3) \quad FR = TR + MPA$$

Another option is to use the current user's profile to estimate the likelihood of him or her being interested in each of the messages in the potential result set. This could be based on which topics of the forum the user is most interested in (this would probably have to be calculated based on usage data).

$$(4) \quad FR = TR + \text{this message's topic's popularity score for this user}$$

This has the risk, as discussed previously, of not finding messages that are posted in wrong topics. A more logical generalisation of this would perhaps be to give preference to messages in the same topic as the question is posted within as in the following:

$$(5) \quad FR = TR + (\text{IF this message's topic} = \text{new question's topic} \\ \text{THEN add bonus})$$

Taking further advantage of the message types does for instance allow comparing the new question to existing questions and returning the answers registered to those questions directly, and not the questions it was compared against. This should be better than simply searching for similarities between the new question and all existing messages as done now. To integrate this with equation 3 would give an algorithm of the form:

$$(6) \quad FR = TR(\text{of searched existing question}) + MPA(\text{for answer associated with this existing question})$$

This would rank all answers associated with the matched questions, based on the textual similarity between the new question and the existing questions and the popularity score for each of the answers of these questions.

Returning answers directly based on question similarity like this is probably the biggest unrealised potential of using the message types for improving searching. Though users might still want to also see the original question, to make sure the answer is what they want to know. Displaying both questions and related answers in the results might be possible, but care must be taken not to make the display too complex.

All of these potential improvements would require testing to see which ones would improve the results and how to best integrate the useful ones into one algorithm.

## 8.2 Further work

In this section possible further ways of extending the system are proposed, in addition to improving the matching accuracy of the Query Monitoring Agent and the display of its search results as identified during evaluation.

For improving the system, a global database with an administrative interface to add and manage forums, servers, and course definitions, would be a logical step in making the system easier to deploy. This could also be exposed as a Web Service that both the Forum Display PHP part and the main Java based Application Layer would access for configuration information limiting their need of each having separate configuration files.

An area that might require more testing and tuning is how well the system scales. Currently it has mostly been tested with two different servers, but as with all distributed systems it is important to find out where bottlenecks are when the load on the system is increased or changed. Two possible solutions

exist if it is found that response times become too slow when accessing the Data Layers of the other instances of the system (the remote servers). Currently the results are retrieved from one server at a time and can be returned without checking all servers if the required result is already found. This performs okay when searching for one record, though response time might be improved somewhat at the expense of requiring more processing and causing more traffic by asking all servers in parallel. But in the cases where all results from all servers for a given query are required it would always make more sense to contact all the servers in parallel instead of serially. This should then improve the system response time without any direct adverse effects. The amount of *work* to be done is the same; the difference would be the added complexity need in the Dispatcher to handle waiting for multiple concurrent connections. Another possible approach would be to cache copies of remote data in the local server, but that raises data consistency issues because data is copied in at least two places and can get out of synchronisation. Therefore it is desirable to avoid this approach.

Other tasks that might improve the system include adding other output options to the Notification module, as mentioned previously, text messages to mobile phones or personalised RSS feeds could be possible extensions. This would make the forum at least partially accessible from mobile devices and also give the users more choice in how they would like to follow the discussions in the forum.

A possible further step would be to allow users to reply directly from the notifications. This assumes a notification contains the full message and that the users would not need to look up other messages in the discussion to reply. In the case of email this could possibly be done by the system having an *inbox* in which it receives messages sent to it as email and then adds them to the system. The email would then need to contain some data readable by the system to identify which message it is a reply to, which user sent it and most problematically the message type of this reply. A possible way of enabling this could be by the system generating custom email addresses for each possible reply action. These could then be presented as a list of *mailto* links at the end

of the notification, for instance “reply to this message with an answer”, “reply to this message with a comment”, etc. These custom email addresses would in that case encode the *id* of the message being replied to, the *id* of the user replying and the *id* of the message type of the reply before the ‘@’ sign in the email address. This instead of the user replying directly to the notification in which case identifying the message type of the reply is difficult. Replying with text messages from mobile phones could also be possible, but how the user would then select the reply type would be a problem. Allowing replies directly from notifications would be a way of further speeding up discussions.

The Query Monitoring Agent could possibly add search results from other normal *unstructured* forums available online, from Wikipedia (Wikipedia, 2006a) or from the Internet in general, to the search results returned after asking a question. These sources would then have to be indexed routinely by a web crawler, or integrated using a third party search engine like Google (Google, 2006). The approach used for ranking the results could also be made more sophisticated, and approaches like Latent Semantic Indexing (Foltz, 1990), Probabilistic Latent Semantic Analysis (Hofmann, 1999) or a modified form of Item Response Theory (Wikipedia, 2006c) are possible improvements. A more active use of the data from the students’ profiles could also be used to customise the search results for each user as discussed in the previous section.

The Tutor Proxy Agent can also be extended with further automated analysis tools to help the teacher track down potential problems in the students’ understanding. This is pointed out as an important way of enhancing current Computer Supported Collaborative Learning environments in a review by Dimitracopoulou (2005). Evaluating the cognitive level at which students are participating as done by Corich, Kinshuk and Hunt (2004) would then be an interesting approach.

Another possible extension is the use of AJAX (Asynchronous JavaScript and XML) technologies in the Presentation Layer of the system, which would make it possible to transfer data directly between the user’s computer and the



Application Layer of the system. In effect, more of the Forum Display would then use JavaScript on the user's computer instead of PHP on the server. One advantage of this approach is, for instance, implementing the possibility to open several messages in full at once, when viewing a list of message subjects. This could also be done with the current architecture, though the tracking of read messages would then have to be more complex, and the AJAX version would be both quicker and more efficient in data traffic than what can be achieved without this approach.

Extending the system to focus more on using discussion forums for discursive learning is another interesting approach. Two particular suggestions have been made in this regard (Moroni, Smestad & Kinshuk, 2006); the first is to allow the users to specify the conversational move they are making with a message in addition to the message type. This can for instance be *agreement*, *disagreement*, *request for clarification* or *giving clarification*. The second suggestion is to provide explicit functions in the system to make summarisation of a discussion easier for the users. This should encourage the users to improve their argumentative strategies and their reflection skills.

In summary the eQuake system was viewed as a successful implementation, and depending on what type of usage scenarios to focus on, various ideas for research into further improvements were presented.



## References

Ackerman, M.S. & McDonald, D.W. (1996), Answer Garden 2: merging organizational memory with collaborative help, in 'CSCW '96: Proceedings of the 1996 ACM conference on Computer supported cooperative work', ACM Press, New York, NY, USA, pp. 97-105.

Apache (2004), 'Apache License', Available from web-site (<http://www.apache.org/licenses>), Accessed: 01. Dec 2006.

Apache (2006), Software package, Available from web-site (<http://httpd.apache.org>), Accessed: 01. Dec 2006.

Apache Axis (2006), Software package, Available from web-site (<http://ws.apache.org/axis>), Accessed: 01. Dec 2006.

Apache Tomcat (2006), Software package, Available from web-site (<http://tomcat.apache.org>), Accessed: 01. Dec 2006.

Avgeriou, P.; Papasalouros, A.; Retalis, S. & Skordalakis, M. (2003), 'Towards a Pattern Language for Learning Management Systems', *Educational Technology & Society* **6**(2), pp. 11-24.

Baraniuk, R.G. (2006), 'Connexions - Building Communities and Sharing Knowledge', Available from web-site (<http://video.google.com/videoplay?docid=6852287090518403675>), Accessed: 30. Oct 2006.

Bell, D. & Grimson, J. (1992), *Distributed Database Systems*, Wokingham, England: Addison-Wesley.

Bielaczyc, K. & Blake, P. (2006), Shifting epistemologies: examining student understanding of new models of knowledge and learning, in 'Proceedings of

the 7th international conference on Learning sciences', pp. 50-56.

BlackBoard (2006), Software package, Available from web-site (<http://www.blackboard.com>), Accessed: 01. Dec 2006.

Bodzin, A.M. & Park, J.C. (2000), 'Dialogue Patterns of Preservice Science Teachers Using Asynchronous Computer-Mediated Communications on the World Wide Web', *Journal of Computers in Mathematics and Science Teaching* **19**(2), pp. 161-194.

Bransford, J.; Brown, A. & Cocking, R., (eds.) (2000), *How People Learn: Brain, Mind, Experience, and School*, National Academy Press, Washington DC.

Brookfield, S.D. (1986), *Understanding and Facilitating Adult Learning*, Jossey-Bass, San Francisco, CA.

Browne, A. (2006), 'E-Learning@Massey', *Off Campus* **11**(4), pp. 10-11.

Calvani, A.; Fini, A.; Pettenati, M.; Masseti, M. & Sarti, L. (2005), 'Collaborative Learning Environments Design: bridging the gap between CSDL theories and Open Source Platforms', *Journal of e-Learning and Knowledge Society, The Italian e-Learning Association Journal* **1**(3).

Caswell, B. & Bielaczyc, K. (2002), 'Knowledge Forum: altering the relationship between students and scientific knowledge', *Education, Communication & Information* **1**(3), pp. 281-305.

Cho, H.; Lee, J.; Stefanone, M. & Gay, G. (2005), 'Development of computer-supported collaborative social networks in a distributed learning community', *Behaviour & Information Technology* **24**(6), pp. 435-447.

Claroline (2006), Software package, Available from web-site (<http://www.claroline.net>), Accessed: 01. Dec 2006.

Corich, S.; Kinshuk & Hunt, L.M. (2004), 'Assessing Discussion Forum Participation: In Search of Quality', *International Journal of Instructional Technology and Distance Learning* **1**(12), pp. 3-12.

CVS (2006), Software package, Available from web-site (<http://www.nongnu.org/cvs>), Accessed: 01. Dec 2006.

Dave, K.; Wattenberg, M. & Muller, M. (2004), Flash forums and forumReader: navigating a new kind of large-scale online discussion, in 'CSCW '04: Proceedings of the 2004 ACM conference on Computer supported cooperative work', ACM Press, New York, NY, USA, pp. 232-241.

Davis, F.D. (1989), 'Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology', *MIS Quarterly* **13**(3), pp. 319-340.

Davis, F.D.; Bagozzi, R.P. & Warshaw, P.R. (1989), 'User Acceptance of Computer Technology: A Comparison of Two Theoretical Models', *Management Science* **35**(8), pp. 982-1003.

Dimitracopoulou, A. (2005), Designing collaborative learning systems: current trends & future research agenda, in 'CSCL '05: Proceedings of the 2005 conference on Computer support for collaborative learning', pp. 115-124.

Donath, J. (2002), 'A semantic approach to visualizing online conversations', *Communications of the ACM* **45**(4), pp. 45-49.

Donath, J.; Karahalios, K. & Viegas, F. (1999), Visualizing Conversation, in 'Thirty-Second Annual Hawaii International Conference on System Sciences', IEEE Computer Society, Los Alamitos, CA, USA, pp. 2023.

Dringus, L.P. & Ellis, T.J. (2004), Building the SCAFFOLD for Evaluating Threaded Discussion Forum Activity: Describing and Categorizing Contributions, in 'FIE 2004: 34th Annual Frontiers in Education, 2004', pp.

T2C- 7-12.

Eclipse (2006), Software package, Available from web-site (<http://www.eclipse.org>), Accessed: 01. Dec 2006.

Eclipse Web Tools Platform (2006), Software package, Available from web-site (<http://www.eclipse.org/webtools>), Accessed: 01. Dec 2006.

Experts Exchange (2006), Web-site (<http://www.experts-exchange.com>), Accessed: 01. Dec 2006.

fabFORCE DBDesigner (2006), Software package, Available from web-site (<http://fabforce.net/dbdesigner4>), Accessed: 01. Dec 2006.

Fini, A. (2006), 'Moodle: Modules and plugins, Activity Module: Forum Plus', Available from web-site (<http://moodle.org/mod/data/view.php?d=13&rid=354>), Accessed: 01. Dec 2006.

Fischer, G. (2000), 'Symmetry of ignorance, social creativity, and meta-design', *Knowledge-Based Systems* **13**(7-8), pp. 527-537.

Fischer, G.; Scharff, E. & Ye, Y. (2004), *Social Capital and Information Technology*, MIT Press, Cambridge, MA, chapter: 'Fostering Social Creativity by Increasing Social Capital', pp. 355-399.

Fishbein, M. & Ajzen, I. (1975), *Belief, Attitude, Intention, and Behavior: An Introduction to Theory and Research*, Addison-Wesley, Reading, MA.

Foltz, P.W. (1990), Using Latent Semantic Indexing for Information Filtering, in R. B. Allen, (ed.), 'Proceedings of the Conference on Office Information Systems', pp. 40-47.

Free Software Foundation (1991), 'GPL – GNU General Public License',

Available from web-site (<http://www.gnu.org/licenses/gpl.txt>), Accessed: 01. Dec 2006.

Freiberger, P. & Swaine, M. (2000), *Fire In The Valley, The Making of the Personal Computer*, McGraw-Hill.

Friesen, N.; Fisher, S. & Roberts, A. (2004), *CanCore Guidelines - for the implementation of Learning Object Metadata (IEEE 1484.12.1-2002)*, Athabasca University.

Google (2006), Web-site (<http://google.com>), Accessed: 01. Dec 2006.

Grieser, G. & Lange, S. (2005), Changing the perspective: Interaction scenarios that take the needs of the machine into account, in 'PISA (Perspectives of Intelligent Systems' Assistance Workshop)', Massey University, Palmerston North.

Hakkarainen, K.; Lipponen, L. & Järvelä, S. (2002), *CSCL 2 : carrying forward the conversation*, Lawrence Erlbaum, Mahwah, NJ, chapter: 'Epistemology of Inquiry and Computer-Supported Collaborative Learning', pp. 129-152.

Heckman, R. & Annabi, H. (2005), 'A Content Analytic Comparison of Learning Processes in Online and Face-to-Face Case Study Discussions', *Journal of Computer-Mediated Communication* **10**(2).

Hibernate (2006), Software package, Available from web-site (<http://www.hibernate.org>), Accessed: 01. Dec 2006.

Hjørland, B. (2002), 'Epistemology and the socio-cognitive perspective in information science', *Journal of the American Society for Information Science and Technology* **53**(4), pp. 257-270.

Hofmann, T. (1999), Probabilistic Latent Semantic Analysis, in 'UAI'99:

Uncertainty in Artificial Intelligence'.

Jonassen, D. & Remidez Jr., H. (2005), 'Mapping alternative discourse structures onto computer conferences', *International Journal of Knowledge and Learning* **1**(1/2), pp. 113-129.

Kaiser, N.C. (2004), 'nPost.com Interview with Ted O'Neill, CEO of InfoPop', Available from web-site (<http://www.npost.com/interview.jsp?intID=INT00067>), Accessed: 24.Nov 2006.

King, W.R. & He, J. (2006), 'A meta-analysis of the technology acceptance model', *Information & Management* **43**(6), pp. 740-755.

Kobielus, J. & Manes, A.T. (2004), 'Service-Oriented Architecture: Developing the Enterprise Roadmap', Technical report, Burton Group.

Lee, J.; Cho, H.; Gay, G.; Davidson, B. & Ingraffea, A. (2003), 'Technology Acceptance and Social Networking in Distance Learning', *Educational Technology & Society* **6**(2), pp. 50-61.

Legrís, P.; Ingham, J. & Collerette, P. (2003), 'Why do people use information technology? A critical review of the technology acceptance model', *Information and Management* **40**(3), pp. 191-204.

Leinonen, T.; Kligyte, G.; Toikkanen, T.; Pietarila, J. & Dean, P. (2003), 'Learning with Collaborative Software - A guide to Fle3', Available from web-site ([http://fle3.uiah.fi/papers/fle3\\_guide.pdf](http://fle3.uiah.fi/papers/fle3_guide.pdf)), Accessed: 03. Nov 2006.

Liu, S.; Liao, H. & Peng, C. (2005), 'Applying the technology acceptance model and flow theory to online e-learning users' acceptance behavior', *Issues in Information Systems* **6**(2), pp. 175-181.

Long, B. & Baecker, R. (1997), A taxonomy of Internet communication tools,

in 'Proceedings of WebNet '97'.

Mazza, R. & Dimitrova, V. (2004), Visualising student tracking data to support instructors in web-based distance education, in 'WWW Alt. '04: Proceedings of the 13th international World Wide Web conference on Alternate track papers & posters', ACM Press, New York, NY, USA, pp. 154-161.

Mazzolini, M. & Maddison, S. (2003), 'Sage, guide or ghost? The effect of instructor intervention on student participation in online discussion forums', *Computers & Education* **40**(3), pp. 237-253.

Microsoft Office (2006), Software package, Available from web-site (<http://office.microsoft.com>), Accessed: 01. Dec 2006.

Microsoft Windows (2006), Software package, Available from web-site (<http://www.microsoft.com/windows>), Accessed: 01. Dec 2006.

MiddlegenIDE (2006), Software package, Available from web-site (<http://boss.bekk.no/boss/middlegen>), Accessed: 01. Dec 2006.

Molnar, A. (1997), 'Computers in Education: A Brief History', *THE Journal (Technological Horizons In Education)* **24**(11).

Moodle (2006), Software package, Available from web-site (<http://moodle.org>), Accessed: 01. Dec 2006.

Mørch, A.; Dolonen, J. & Omdahl, K. (2003), Integrating Agents with an Open Source Learning Environment, in 'Proceedings of International Conference on Computers in Education'.

Mørch, A.I.; Dolonen, J.A. & Nævdal, J.E.B. (2006), 'An evolutionary approach to prototyping pedagogical agents: from simulation to integrated system', *Journal of Network and Computer Applications* **29**(2-3), pp. 177-



Moroni, C.; Smestad, O. & Kinshuk (2006), Improving discursive negotiation in web based discussion forums, in 'Cognition and Exploratory Learning in Digital Age (CELDA 2006)'.

Muirhead, B. (2000), 'Enhancing Social Interaction in Computer-Mediated Distance Education', *Educational Technology & Society* **3**(4), pp. 1-11.

MySQL (2006), Software package, Available from web-site (<http://www.mysql.com>), Accessed: 01. Dec 2006.

MySQL Workbench (2006), Software package, Available from web-site (<http://www.mysql.com/products/tools/workbench>), Accessed: 01. Dec 2006.

Newman, P.S. (2001), 'Treetables and Other Visualizations for Email Threads', Technical report, Xerox PARC.

Ng'ambi, D. & Hardman, J. (2004), 'Towards a knowledge-sharing scaffolding environment based on learners' questions', *British Journal of Educational Technology* **35**(2), pp. 187-196.

Nijholt, A. (2002), Computer-facilitated Community Building for E-Learning: Panel Introduction, in 'Proceedings of The IEEE International Conference on Advanced Learning Technologies, 2002', pp. 541-544.

OSNews (2006), Web-site (<http://osnews.com>), Accessed: 01. Dec 2006.

Paulsen, M.F. (2003), 'Experiences with Learning Management Systems in 113 European Institutions', *Educational Technology & Society* **6**(4), pp. 134-148.

Peak, P. (2005), 'Hibernate vs. Rails: The Persistence Showdown', Available

from web-site  
(<http://www.theserverside.com/tt/articles/article.tss?l=RailsHibernate>),  
Accessed: 05. Oct 2006.

phpBB (2006), Software package, Available from web-site  
(<http://www.phpbb.com>), Accessed: 01. Dec 2006.

PHPeclipse (2006), Software package, Available from web-site  
(<http://www.phpclipse.net>), Accessed: 01. Dec 2006.

phpMyAdmin (2006), Software package, Available from web-site  
(<http://www.phpmyadmin.net>), Accessed: 01. Dec 2006.

Pike, R.; Dorward, S.; Griesemer, R. & Quinlan, S. (2005), 'Interpreting the Data: Parallel Analysis with Sawzall', *Scientific Programming Journal* **13**(4), pp. 277-298.

Pilkington, R.; Bennett, C. & Vaughan, S. (2000), 'An Evaluation of Computer Mediated Communication to Support Group Discussion in Continuing Education', *Educational Technology & Society* **3**(3), pp. 349-360.

Popolov, D.; Callaghan, M. & Luker, P. (2000), Conversation space: visualising multi-threaded conversation, in 'AVI '00: Proceedings of the working conference on Advanced visual interfaces', ACM Press, New York, NY, USA, pp. 246-249.

Putty (2006), Software package, Available from web-site  
(<http://www.chiark.greenend.org.uk/~sgtatham/putty>), Accessed: 01. Dec 2006.

Rahikainen, M.; Järvelä, S. & Salovaara, H. (2000), Motivational Processes in CSILE-Based Learning, in B. Fishman & S. O'Connor-Divelbiss, (ed.), 'Fourth International Conference of the Learning Sciences', Erlbaum, Mahwah, NJ, pp. 50-51.

Red Hat (2006), Software package, Available from web-site (<http://www.redhat.com/rhel>), Accessed: 01. Dec 2006.

Reichling, T.; Schubert, K. & Wulf, V. (2005), Matching human actors based on their texts: design and evaluation of an instance of the ExpertFinding framework, in 'GROUP '05: Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work', ACM Press, New York, NY, USA, pp. 61-70.

Salton, G. & Buckley, C. (1988), 'Term-weighting approaches in automatic text retrieval', *Information Processing and Management: an International Journal* **24**(5), pp. 513-523.

Sakai (2006), Software package, Available from web-site (<http://sakaiproject.org>), Accessed: 01. Dec 2006.

Scardamalia, M. & Bereiter, C. (1993), 'Technologies for knowledge-building discourse', *Communications of the ACM* **36**(5), pp. 37-41.

Scardamalia, M. & Bereiter, C. (1994), 'Computer Support for Knowledge-Building Communities', *The Journal of the Learning Sciences* **3**(3), pp. 265-283.

Scardamalia, M. & Bereiter, C. (1996), 'Student communities for the advancement of knowledge', *Communications of the ACM* **39**(4), pp. 36-37.

Scardamalia, M. & Bereiter, C. (2005), 'Does education for the knowledge age need a new science?', *European Journal of School Psychology* **3**(1), pp. 21-40.

Scardamalia, M. & Bereiter, C. (2006), *Cambridge Handbook of the Learning Sciences*, Cambridge University Press, chapter: 'Knowledge Building: Theory, Pedagogy, and Technology'.

Scardamalia, M. (2002), *Liberal education in a knowledge society*, Open Court, Chicago, chapter: 'Collective cognitive responsibility for the advancement of knowledge', pp. 76-98.

Schacter, J. (1999), 'The Impact of Education Technology on Student Achievement', Technical report, The Milken Exchange on Education Technology, Santa Monica, CA.

Shadow netWorkspace (2006), Software package, Available from web-site (<http://sns.internetschools.org>), Accessed: 01. Dec 2006.

Slashdot (2006), Web-site (<http://slashdot.org>), Accessed: 01. Dec 2006.

Slavin, R.E. (1983), *Cooperative learning*, Longman, New York.

Subclipse (2006), Software package, Available from web-site (<http://subclipse.tigris.org>), Accessed: 01. Dec 2006.

Subversion (2006), Software package, Available from web-site (<http://subversion.tigris.org>), Accessed: 01. Dec 2006.

Sun (2006), Press release, Available from web-site (<http://www.sun.com/smi/Press/sunflash/2006-11/sunflash.20061113.1.xml>), Accessed: 26. Nov 2006.

Sun JDK (2006), Software package, Available from web-site (<http://java.sun.com/javase>), Accessed: 01. Dec 2006.

Tretiakov, A.; Smestad, O. & Kinshuk (2006), Improving the quality of discussion forum discourses by using an internal market, in 'IEEE International Conference on Advanced Learning Technologies (ICALT 2006)', pp. 601-602.

Vivacqua, A. & Lieberman, H. (2000), Agents to assist in finding help, in 'CHI

'00: Proceedings of the SIGCHI conference on Human factors in computing systems', ACM Press, New York, NY, USA, pp. 65-72.

W3C (2006), Web Services, Web-site (<http://www.w3.org/2002/ws>), Accessed: 01. Dec 2006.

Watson, G. & Prestridge, S. (2003), 'A networked learning community approach to sustain teacher ICT professional development', *Australian Journal of Educational Technology* **19**(2), pp. 227-240.

WebCT (2006), Software package, Available from web-site (<http://www.webct.com>), Accessed: 01. Dec 2006.

Weller, M. & Whitsed, N. (2004), 'Learning and teaching - Developing pedagogy for the Internet', *Health Information and Libraries Journal* **21**(1), pp. 74-77.

Wikipedia (2006a), Web-site (<http://wikipedia.org>), Accessed: 01. Dec 2006.

Wikipedia (2006b), "Wiki", Available from web-site (<http://en.wikipedia.org/w/index.php?title=Wiki&oldid=89448131>) Accessed: 24. Nov 2006.

Wikipedia (2006c), "Item response theory", Available from web-site ([http://en.wikipedia.org/w/index.php?title=Item\\_response\\_theory&oldid=88811455](http://en.wikipedia.org/w/index.php?title=Item_response_theory&oldid=88811455)) Accessed: 24. Nov 2006.

WinSCP (2006), Software package, Available from web-site (<http://winscp.net>), Accessed: 01. Dec 2006.

Winter, E.C. & McGhie-Richmond, D. (2005), 'Using computer conferencing and case studies to enable collaboration between expert and novice teachers', *Journal of Computer Assisted Learning* **21**(2), pp. 118-129.

Wooldridge, M. (1999), 'Diversity and agent technology', *The Knowledge Engineering Review* **14**(2), pp. 151-152.

XFree86 (1996), 'BSD – (Modified) Berkeley Software Distribution License', Available from web-site (<http://www.xfree86.org/3.3.6/COPYRIGHT2.html#5>), Accessed: 01. Dec 2006.

Yahoo! Groups (2006), Web-site (<http://groups.yahoo.com>), Accessed: 01. Dec 2006.

Zhang, Y.; Kinshuk; Smestad, Ø.; Yang, J. & Jeffery, L. (2006), Using Agents for Enhancing Learning Effects in an Advanced Discussion Forum, in 'Proceedings of International Conference on Computers in Education, 2006'.

## Appendix A: Data definitions

### A.1 Global data section:

**equake\_post\_type:** Lists the possible types of content a message can contain, for instance, question, answer or comment. Also keeps track of which types can only be used by tutors (FAQs) and which types can or cannot be used to start a new discussion, for instance a discussion cannot be started with an answer.

**equake\_scale:** It defines scales used by students for rating the messages.

**equake:** This is the highest scope table, which lists the forums available and the scale they use for ratings.

**equake\_course:** This table defines that a course is participating in a specific forum, and that it resides on a given server.

**equake\_server:** It contains information about a server an instance of the system is running on, like URLs and information on the institution this server belongs to.

### A.2 Local data section:

**equake\_post\_keywords:** It contains the keywords of a message as used by the Query Monitoring Agent.

**equake\_post:** It contains all messages and associated data made by students and teachers in courses on a server.

**equake\_discussion:** It has the discussion topics in a forum defined by the teachers.



**equake\_wants\_notification:** It tracks which users are subscribed to notifications of new messages: in reply to a given message, in a specific topic or in a forum.

**equake\_post\_rating:** It has one entry containing the rating given for each message a student has rated. This makes sure that each student only has one *vote* in calculating a message's rating, while allowing the students to change their rating for any message as many times as they wish.

**equake\_post\_read:** It has one entry for each time a user views a given message. This will have multiple entries for one message and one user when the user reads a message more than once.

**equake\_group:** It contains study groups that have been defined by teachers or students.

**equake\_discussion\_tutor:** It defines that a tutor or teacher is responsible for moderating in a given discussion topic. Several tutors can be moderators in one topic (and a tutor can be a moderator in several topics).

**equake\_user:** It contains all user information, most of which is imported from the LMS if the user has specified it there. This also maps the internal *userid* of eQuake to the id of the user in the LMS through *useridinlms*.

**equake\_group\_member:** It contains entries for each student that is part of a given study group and how competent the student feels he or she is in the subject area of the study group.

**equake\_tutor\_specialty:** This contains keywords tutors have registered as their specialty.

**equake\_course\_tutor:** This user is registered as a tutor in the given course.

**equake\_course\_student:** This user is registered as a student in the given course. This table also keeps track of any tutor assigned to a student and the number of messages the student has submitted within this forum.

**equake\_discussion\_profile:** This table keeps track of how active a student has been within the different discussion topics within a forum. This data could instead be calculated on the fly, but is stored to save some calculations and to simplify some SQL statements.

# Appendix B: Evaluation forms

## B.1 eQuake evaluation questionnaire for users

For each question listed below, cross the number on the scale that best fits your judgment.

*Thank you for your participation.*



**Having examined/tested eQuake, I intend to use it in my degree program.**

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

**In the future, if I have access to eQuake, I would use it.**

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

**I find eQuake useful in my degree program.**

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

**Using eQuake would:**

a) Improve my performance in my degree program.

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

b) Increase my productivity<sup>3</sup> in my degree program.

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

c) Enhance my effectiveness<sup>4</sup> in my degree program.

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

**In relation to posting messages in the forum:**

a) Interacting with eQuake does not require a lot of mental effort from me.

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

b) I find eQuake easy to use.

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

c) I find it easy to get eQuake to do what I want to do.

Strongly agree	+3	+2	+1	0	- 1	-2	-3	Strongly disagree

<sup>3</sup> *Productivity*: the system enables you to save time, work, etc.

<sup>4</sup> *Effectiveness*: the system improves your capability of achieving the goal of getting your degree.

*In relation to finding an answer to a posted question:*

a) Interacting with eQuake does not require a lot of mental effort by me.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

b) I find eQuake easy to use.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

c) I find it easy to get eQuake to do what I want to do.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

*In relation to rating of answer features:*

a) Interacting with eQuake does not require a lot of mental effort by me.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

b) I find eQuake easy to use.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

c) I find it easy to get eQuake to do what I want to do.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

*In relation to subscription features:*

a) Interacting with eQuake does not require a lot of mental effort by me.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

b) I find eQuake easy to use.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

c) I find it easy to get eQuake to do what I want to do.

Strongly agree	+3	+2	+1	0	-1	-2	-3	Strongly disagree

**USAGE**

Indicate how often you used eQuake.

frequently	more than once a day	about once a day	4 or 6 times a week	2 or 3 time a week	about once a week	less than once a week	not at all	infrequently

**COMMENTS**

Please feel free to add any comments relating to the use and features of eQuake that have not been addressed by this survey.

The evaluation of the system is being conducted by Mr Oyvind Smestad ( [o.smestad@massey.ac.nz](mailto:o.smestad@massey.ac.nz) ). "This project has been evaluated by peer review and judged to be low risk. Consequently, it has not been reviewed by one of the University's Human Ethics Committees. The researcher(s)named above are responsible for the ethical conduct of this research. If you have any concerns about the conduct of this research that you wish to raise with someone other than the researcher(s), please contact Professor Sylvia Rumball, Assistant to the Vice-Chancellor (Ethics & Equity), telephone 06 350 5249, e-mail [humanethicspn@massey.ac.nz](mailto:humanethicspn@massey.ac.nz)".

## B.2 eQuake evaluation questionnaire for staff

### eQuake Evaluation Form

Course: \_\_\_\_\_ Faculty/University: \_\_\_\_\_ Date: \_\_\_\_\_

For each question listed below, circle the number on the Scale that best fits your judgment. Thank you for your participation.

1 = Strongly disagree

5 = Strongly agree

NA= Not applicable

Description of Item	Scale					
<b>Basic infrastructure of the system</b>						
1.1 The system provides an excellent student-student interaction. -----	1	2	3	4	5	NA
1.2 The system provides an excellent student-teacher (or vice-versa) interaction. -----	1	2	3	4	5	NA
1.3 The system provides an excellent environment for posting queries and solutions -----	1	2	3	4	5	NA
1.4 The system flags for significant and frequently repeated queries. -----	1	2	3	4	5	NA
1.5 The system allows a teacher to intervene by creating FAQs. -----	1	2	3	4	5	NA
<b>Query display component</b>						
2.1 The system allows users to view queries. -----	1	2	3	4	5	NA
2.2 The system allows users to search queries. -----	1	2	3	4	5	NA
2.3 The system allows users to retrieve queries. -----	1	2	3	4	5	NA
2.4 Overall, the query display system is easy to use and navigate. -----	1	2	3	4	5	NA
<b>Student proxy agent</b>						
3.1 The system allows students to post queries. -----	1	2	3	4	5	NA
3.2 The system allows students to post comments and/or solutions. -----	1	2	3	4	5	NA
3.3 The system allows students to rate the responses of others to their queries. -----	1	2	3	4	5	NA
3.4 The system forwards teacher formatted answer of a topic to designated students. -----	1	2	3	4	5	NA
3.5 The system identifies students who rarely participate or exhibit lack of understanding of the already existing solutions. -----	1	2	3	4	5	NA
3.6 Overall, the student proxy agent is easy to use and navigate -----	1	2	3	4	5	NA
<b>Query monitoring agent</b>						
4.1 The system detects queries pertaining to a question/topic that had been posted. -----	1	2	3	4	5	NA
4.2 The system flags for the frequently repeated queries on the same question/topic and send it with all the related postings to the teacher through the tutor proxy agent. -----	1	2	3	4	5	NA
4.3 The system is easy to configure and queries/postings can either be send to all the teachers (who are involved with the subject area) or only to the teacher whose students had asked majority of the questions on the topic. -----	1	2	3	4	5	NA
4.4 Overall, the query monitoring agent is easy to use and navigate. -----	1	2	3	4	5	NA

<b>Tutor proxy agent and target selection agent</b>						
5.1	The tutor proxy agent is easy to use and navigate. -----	1	2	3	4	5 NA
5.2	Tutor proxy agent is robust (performed well) -----	1	2	3	4	5 NA
5.3	The target selection agent is easy to use and navigate -----	1	2	3	4	5 NA
5.4	The target selection agent is robust (performed well) -----	1	2	3	4	5 NA
<b>Learning community</b>						
6.1	The system supports Massey University learning community very well. -----	1	2	3	4	5 NA
6.2	The system supports EIT learning community very well -----	1	2	3	4	5 NA
6.3	The system supports other learning community very well -----	1	2	3	4	5 NA
<b>General – usability testing</b>						
7.1	On the first impression, the eQuake site looks very good (Not very cluttered) -----	1	2	3	4	5 NA
7.2	Navigation features are easy to identify and use -----	1	2	3	4	5 NA
7.3	Background colours do not hinder readability -----	1	2	3	4	5 NA
7.4	Text style and font sizes are appropriate -----	1	2	3	4	5 NA
7.5	All the links I clicked worked -----	1	2	3	4	5 NA
7.6	All pages were loaded reasonably fast -----	1	2	3	4	5 NA

1 = Strongly disagree

5 = Strongly agree

NA= Not applicable

Please put in the space provided any helpful suggestions for the improvement of eQuake that you wish to make.

Comments:



## Appendix C: Evaluation results

### C.1 User evaluation results

Variable	Item	Mean	Std. Deviation
Behavioural intention (BI)	Having examined/tested eQuake, I intend to use it in my degree program.	1.34	1.327
	In the future, if I have access to eQuake, I would use it.	1.83	1.200
Perceived Usefulness (U)	I find eQuake useful in my degree program.	1.20	1.132
	Using eQuake would improve my performance in my degree program.	1.46	.980
	Using eQuake would increase my productivity in my degree program.	1.3429	1.30481
	Using eQuake would enhance my effectiveness in my degree program.	1.54	1.379
Perceived Ease of use in relation to posting a message.  EOU(posting)	In relation to posting a message in the forum: interacting with eQuake does not require a lot of mental effort from me.	1.43	1.399
	In relation to posting a message in the forum: I find eQuake easy to use.	1.5429	1.52128
	In relation to posting a message in the forum: I find easy to get eQuake to do what I want to do	1.5429	1.29121
Perceived Ease of Use in relation to finding an answer.  EOU (finding)	In relation to finding an answer: interacting with eQuake does not require a lot of mental effort from me	1.2000	1.54919
	In relation to finding an answer: I find eQuake easy to use	1.4000	1.80196
	In relation to finding an answer: I find easy to get eQuake to do what I want to do	1.0571	1.57074
Perceived Ease of Use in relation to rating of answer.  EOU (rating)	In relation to rating of answer: interacting with eQuake Does not require a lot of mental effort from me	1.7429	1.33599
	In relation to rating of answer: I find eQuake easy to use	1.7429	1.29121
	In relation to rating of answer: I find easy to get eQuake to do what I want to do	1.3143	1.47072
Perceived Ease of Use in relation to subscription features.  EOU (subscription)	In relation to subscription features: interacting with eQuake Does not require a lot of mental effort from me	1.5429	1.48211
	In relation to subscription features: I find eQuake easy to use	1.6571	1.39205
	In relation to subscription features: I find easy to get eQuake to do what I want to do	1.6571	1.30481

## C.2 Teacher evaluation results

	Question	Satisfaction level
<b>Basic infrastructure</b>	1.1 The system provides an excellent student-student interaction.	87.5%
	1.2 The system provides an excellent student-teacher (or vice-versa) interaction.	87.5%
	1.3 The system provides an excellent environment for posting queries and solutions	87.5%
	1.4 The system flags for significant and frequently repeated queries.	75%
	1.5 The system allows a teacher to intervene by creating FAQs.	100%
<b>Query display component</b>	2.1 The system allows users to view queries.	100%
	2.2 The system allows users to search queries.	62.5%
	2.3 The system allows users to retrieve queries.	62.5%
	2.4 Overall, the query display system is easy to use and navigate	75%
<b>Student proxy agent</b>	3.1 The system allows students to post queries.	100%
	3.2 The system allows students to post comments and/or solutions.	100%
	3.3 The system allows students to rate the responses of others to their queries.	87.5%
	3.4 The system forwards teacher formatted answer of a topic to designated students.	75%
	3.5 The system identifies students who rarely participate or exhibit lack of understanding of the already existing solutions.	75%
	3.6 Overall, the student proxy agent is easy to use and navigate	100%
<b>Query monitoring agent</b>	4.1 The system detects queries pertaining to a question/topic that had been posted.	50%
	4.2 The system flags for the frequently repeated queries on the same question/topic and send it with all the related postings to the teacher through the tutor proxy agent.	37.5%
	4.3 The system is easy to configure and queries/postings can either be send to all the teachers (who are involved with the subject area) or only to the teacher whose students had asked majority of the questions on the topic.	37.5%
	4.4 Overall, the query monitoring agent is easy to use and navigate.	50%
<b>Tutor proxy agent and target selection agent</b>	5.1 The tutor proxy agent is easy to use and navigate.	87.5%
	5.2 The Tutor proxy agent is robust (performed well)	75%
	5.3 The target selection agent is easy to use and navigate	75%
	5.4 The target selection agent is robust (performed well)	75%
<b>Learning community</b>	6.1 The system supports Massey University learning community very well.	25%
	6.2 The system supports EIT learning community very well	75%
	6.3 The system supports other learning community very well	75%
<b>General usability</b>	7.1 On the first impression, the eQuake site looks very good (Not very cluttered)	100%
	7.2 Navigation features are easy to identify and use	87.5%
	7.3 Background colours do not hinder readability	100%
	7.4 Text style and font sizes are appropriate	87.5%
	7.5 All the links I clicked worked	100%
	7.6 All pages were loaded reasonably fast	100%

## Appendix D: List of terms

AI – Artificial Intelligence  
AJAX – Asynchronous JavaScript and XML  
ALTRC – Advanced Learning Technologies Research Centre  
API – Application Programming Interface  
AUT – Auckland University of Technology  
BBS – Bulletin Board System  
BI – Behavioural Intention to use (part of Technology Acceptance Model)  
BSD –Berkeley Software Distribution  
CMS – Course Management System  
CSCL – Computer Supported Collaborative Learning  
CSILE – Computer-Supported Intentional Learning Environment  
DB – Database  
DFAQ – Dynamic Frequently Asked Questions environment  
EIT – Eastern Institute of Technology  
EOU – Perceived Ease Of Use (part of Technology Acceptance Model)  
eQuake – electronic Question and Answer Knowledge Environment  
ER – Entity Relationship  
FAQ – Frequently Asked Question  
FLE – Future Learning Environment  
FR – Final Ranking score  
GNU – GNU's Not UNIX  
GPL – GNU General Public License  
GUI – Graphical User Interface  
HTML – HyperText Markup Language  
HTTP – HyperText Transfer Protocol  
I/O – Input / Output  
IDE – Integrated Development Environment  
IS – Information Systems  
IT – Information technology  
JDBC – Java Database Connectivity  
JDK – Java Development Kit

JSP – JavaServer Pages  
LMS – Learning Management System  
LSI – Latent Semantic Indexing  
MPA – Message Popularity All users  
MPC – Message Popularity Class users  
NA – Not Applicable  
ORM – Object Relational Mapping  
PA – Proxy Agent (part of eQuake system)  
PC – Personal Computer  
PHP – PHP: Hypertext Preprocessor  
QMA – Query Monitoring Agent (part of eQuake system)  
RSS – Really Simple Syndication  
SDK – Software Development Kit  
SNS – Shadow netWorkspace  
SOA – Service-Oriented Architecture  
SPA – Student Proxy Agent (part of eQuake system)  
SQL – Structured Query Language  
TAM – Technology Acceptance Model  
TPA – Tutor Proxy Agent (part of eQuake system)  
TR – Textual search algorithm Ranking  
TRA – Theory of Reasoned Action  
TSA – Target Selection Agent (part of eQuake system)  
U – Perceived Usefulness (part of Technology Acceptance Model)  
UBB – Ultimate Bulletin Board  
URL – Uniform Resource Locator  
VLE – Virtual Learning Environment  
WS – Web Service  
WSDL – Web Services Description Language  
WTP – Web Tools Platform (part of Eclipse)  
XHTML – Extensible HyperText Markup Language  
XML – eXtensible Markup Language