

# **Combining Computer-Supported, Collaborative Learning with E-Assessment**

## **Enhancing a Wiki System with Flexible Assessment Methods**

**Master Thesis**

at

University of Technology, Graz

submitted by:

Dominik Kowald

**Supervisor:**

Univ.-Doz. Dipl.-Ing. Dr.techn. Christian Gütl

IICM, University of Technology Graz

**Co-Supervisor:**

M.Sc. Mohammad AL-Smadi

IICM, University of Technology Graz

**Institute for Information Systems and Computer Media (IICM)**

University of Technology, Graz

A-8010 Graz, Austria



Graz, May 7, 2012

# **Kombination von Computergestützten, Kollaborativen Lernens mit E-Assessment**

**Erweiterung eines Wiki Systems mit Flexiblen  
Assessment Methoden**

## **Diplomarbeit**

an der

Technischen Universität, Graz

verfasst von:

Dominik Kowald

### **Betreuer:**

Univ.-Doz. Dipl.-Ing. Dr.techn. Christian Gütl

IICM, Technische Universität Graz

### **Mitbetreuer:**

M.Sc. Mohammad AL-Smadi

IICM, Technische Universität Graz

Institut für Informationssysteme und Neue Medien (IICM)

Technische Universität, Graz

A-8010 Graz, Österreich



Graz, Mai 7, 2012

# **Abstract**

The increasing number of students each year has significantly increased the need for effective learning methods that also support the work in groups. This is necessary because the number of teachers and tutors usually stays the same which makes it impossible that typical classroom exercises still can be administrated as individual work. In addition, nowadays the job market asks for employees that work well in a team and so collaborative skills should also be practiced in educational settings such as schools or universities. Fortunately, the development of modern information and communication technologies (ICT) and the emergence of Web 2.0, also referred to as the social Web, support these needs with powerful tools for collaborative learning and writing such as Wikis or Blogs. However, these tools lack of motivational aspects such as assessment and feedback processes which should be an integral part of every learning activity.

Due to these findings, the aim of this thesis is to analyze how computer-supported learning environments can be combined with E-Assessment methods and other functionalities that support teachers and students in their learning and assessment activities. Thus, a Wiki system has been enhanced and integrated in Learning Management Systems (LMS) in order to ensure task and social awareness and to increase motivational aspects. The developed Wiki prototype has also been used in different experimentations to test the functionalities and to get feedback for possible improvements. The experimentation results have shown that the developed forms for self, peer and group assessment have motivated the students and supported them in coordinating with their group members.

# Kurzfassung

Die zunehmende Anzahl an Studenten hat den Bedarf an effektiven Lernmethoden, welche das Arbeiten in Gruppen unterstützen, stark erhöht. Dies ist unumgänglich, da die Anzahl an Professoren und Tuto ren konstant bleibt und es somit unmöglich macht typische Vorlesungsübungen weiterhin als Einzelarbeiten durchzuführen. Weiters ist Teamfähigkeit sehr wichtig um am heutigen Arbeitsmarkt bestehen zu können. Dies unterstreicht die Notwendigkeit an Übungen an Universitäten und Schulen, welche die kollaborativen Fähigkeiten verbessern. Glücklicherweise wird dieses Vorhaben durch die Entwicklung von modernen Informations- und Kommunikationssystemen und dem Aufstieg des Web 2.0, auch bekannt als soziales Web, unterstützt. Diese Technologien bieten mächtige Anwendungen für kollaboratives Lernen und Schreiben, wie Wikis oder Blogs, an. Auf der anderen Seite fehlt es diesen Anwendungen üblicherweise an Motivationsaspekten wie Assessment und Feedback Aktivitäten, welche allerdings Bestandteil eines jeden Lernprozesses sein sollten.

Auf Grund dieser Tatsachen ist es das Ziel dieser Masterarbeit zu analysieren wie computerunterstützte Lernumgebungen mit E-Assessment Methoden und anderen Funktionalitäten, welche Professoren und Studenten in deren Lern- und Assessmentaktivitäten unterstützen, verbunden werden können. Demzufolge wurde ein Wiki System erweitert und in bestehende Lernplattformen integriert um das Aufgabenbewusstsein und die Motivation von Benutzern zu erhöhen. Dieses Wiki System wurde weiters in mehreren Experimenten verwenden um die Funktionalitäten zu testen und Feedback von Benutzern zu bekommen. Dieses Feedback wurde in weiterer Folge zur Entwicklung von neuen Funktionalitäten benutzt. Die Ergebnisse dieser Experimente haben gezeigt, dass die Studenten durch die integrierten Formulare für Self-, Peer- und Group-Assessment in ihrer Arbeit motiviert wurden. Darüber hinaus unterstützten diese Aktivitäten die Koordinierung der Studenten innerhalb ihrer Gruppen.

## **STATUTORY DECLARATION**

I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.

Graz, May 7, 2012

.....  
date

.....  
(signature)

# Notes of Thanks

I want to thank my parents, my colleagues and my close friends in supporting me during all my years in university, especially during the work on this thesis. Furthermore, I want to particularly thank my supervisor Univ.-Doz. Dipl.-Ing. Dr.techn. Christian Gütl for supporting me in any way and for giving me the possibility to be an active part of his research group since my bachelor thesis. Another big thank you is for my co-supervisor M.Sc. Mohammad AL-Smadi who supported me during the practical parts of my work since my master project. I am also very thankful for the tips and help from the rest of our research group, especially Mag. Dr. Gudrun Wesiak, Mag. Phil. Isabella Pichlmair and Joachim Maderer.

Many thanks to all of you!

# Contents

<b>1</b>	<b>Introduction .....</b>	<b>1</b>
1.1	Motivation and Objective.....	1
1.2	Structure of the Work .....	2
<b>2</b>	<b>Technology Enhanced Learning .....</b>	<b>3</b>
2.1	Introduction to Technology Enhanced Learning .....	3
2.2	E-Learning .....	4
2.3	Approaches of Technology Enhanced Learning .....	5
2.4	Summary.....	6
<b>3</b>	<b>Computer-Supported, Collaborative Learning .....</b>	<b>7</b>
3.1	Introduction to Computer-Supported, Collaborative Learning.....	7
3.1.1	Definition and Aspects of Computer-Supported, Collaborative Learning....	7
3.1.2	Motivation and Principles for Supporting Collaborative Learning with Modern Technologies .....	8
3.1.3	History of Computer-Supported, Collaborative Learning .....	9
3.2	Collaborative Learning Tools .....	9
3.2.1	E-tivities.....	10
3.2.2	Types of Collaborative Software and Learning Tools .....	11
3.3	Computer-Supported, Collaborative Writing .....	12
3.3.1	Process and Complexity of Collaborative Writing .....	13
3.3.2	Effects of CSCL on Collaborative Writing.....	13
3.4	Wiki Systems.....	14
3.4.1	Definition and Characteristics of Wiki Systems.....	15
3.4.2	Wikis in Education .....	15
3.5	Summary.....	19
<b>4</b>	<b>E-Assessment .....</b>	<b>21</b>
4.1	Introduction to E-Assessment.....	21
4.1.1	Definition and History of E-Assessment.....	21
4.1.2	Motivations for the Use of E-Assessment .....	22
4.2	How to Differentiate Assessment Forms?.....	23

4.2.1	Strategies of Assessment .....	23
4.2.2	Types of Assessment.....	25
4.3	Assessment and Computer Science .....	26
4.3.1	E-Assessment Types .....	26
4.3.2	Computer-Based versus Paper-Based Assessment.....	27
4.4	Assessment Rubrics.....	28
4.5	Summary.....	30
<b>5</b>	<b>Combining Computer-Supported, Collaborative Learning with E-Assessment .....</b>	<b>32</b>
5.1	Motivation and Problem Definition for Using E-Assessment Together with CSCL .....	32
5.2	Related Work .....	33
5.2.1	Assessing Individual Contributions in Collaborative Writing Projects.....	34
5.2.2	Using Rubrics and Peer Assessment in Wiki Systems .....	36
5.3	Identified Issues.....	38
5.4	Solution Approaches.....	39
5.5	Summary.....	40
<b>6</b>	<b>Requirements and Design of the Prototype.....</b>	<b>42</b>
6.1	Project Idea and Description .....	42
6.2	Requirements .....	43
6.2.1	Functional Requirements .....	44
6.2.2	Non-Functional Requirements.....	48
6.3	Conceptual Architecture.....	49
6.4	Considerations Regarding the Wiki Software .....	51
6.5	Technical Aspects of ScrewTurn Wiki.....	52
6.5.1	System Design.....	52
6.5.2	Wiki Providers .....	56
6.6	Design Considerations.....	57
6.6.1	Data Representation of the Assignment Management .....	57
6.6.2	Assessment and Feedback Cycle .....	58
6.7	Summary.....	59
<b>7</b>	<b>Development of the Prototype.....</b>	<b>61</b>
7.1	Used Technologies, Frameworks and Tools.....	61
7.1.1	Technologies .....	61

7.1.2	Frameworks .....	63
7.1.3	Tools.....	64
7.2	Implementation of the Standalone Prototype .....	65
7.2.1	System Overview.....	65
7.2.2	Data Model.....	66
7.2.3	Programming Documentation .....	69
7.3	Integration of the Prototype .....	74
7.3.1	Login via SSO .....	75
7.3.2	Assignment Authoring Web Services.....	77
7.3.3	Grading Schema.....	78
7.4	Summary.....	79
<b>8</b>	<b>Usage Point of View and Experimentations .....</b>	<b>81</b>
8.1	Installation and Setup.....	81
8.2	Wiki Functionalities.....	82
8.2.1	Assignment Authoring .....	82
8.2.2	Assessment and Feedback Activities.....	84
8.2.3	Student Views .....	87
8.2.4	Teacher Views.....	89
8.3	Experimentations .....	91
8.3.1	Experimentations Overview .....	92
8.3.2	Results and Findings .....	93
8.4	Summary.....	95
<b>9</b>	<b>Lessons Learned .....</b>	<b>97</b>
9.1	Personal Point of View .....	97
9.2	Technical Point of View .....	97
9.3	Usage Point of View.....	98
<b>10</b>	<b>Conclusion.....</b>	<b>99</b>
10.1	Summary.....	99
10.2	Future Work .....	100
<b>11</b>	<b>Bibliography .....</b>	<b>101</b>
<b>12</b>	<b>Appendix: CD-ROM .....</b>	<b>107</b>

# List of Figures

Figure 1: Five-Step Model to create E-tivities (Salmon, 2002).....	10
Figure 2: Quadrants to show the dimensions of CSCL and CSCW (Rama & Bishop, 2006) .....	11
Figure 3: Strategies of assessment (Crisp, 2009) .....	24
Figure 4: Relationships between CBA, CAA, CBL and CAL (Higgins & Bligh, 2006)....	26
Figure 5: Example assessment rubric for paper abstracts (TLT, 2009) .....	29
Figure 6: Example of a contribution summary graph (Biuk-Aghai, Kelen, & Venkatesan, 2008) .....	34
Figure 7: Example of an analysis graph (Biuk-Aghai, Kelen, & Venkatesan, 2008).....	35
Figure 8: Example of a text author graph (Biuk-Aghai, Kelen, & Venkatesan, 2008) .....	35
Figure 9: Example rubric for assessment using Wikis (Jones, Kolloff, & Kolloff, 2010)....	37
Figure 10: Conceptual architecture (AL-Smadi, Guetl, Caballe, & Dunwell, 2012).....	50
Figure 11: ScrewTurn Wiki system design .....	53
Figure 12: Wiki providers management screen (ScrewTurn, 2012) .....	56
Figure 13: Hierarchical data representation of the Wiki .....	57
Figure 14: System overview of the standalone prototype .....	66
Figure 15: Assessment side of the database model.....	67
Figure 16: Assignment side of the database model.....	68
Figure 17: Interactions of the repository pattern (MSDN, 2012) .....	69
Figure 18: Example of the Wiki contribution visualization.....	71
Figure 19: Assessment control used for self assessment .....	71
Figure 20: Example of the graph control to visualize student contributions.....	72
Figure 21: Sequence diagram of the SSO login process.....	76
Figure 22: Sequence diagram of the assignment authoring process .....	77
Figure 23: Sequence diagram of the grading process.....	78
Figure 24: Assignment authoring .....	82
Figure 25: Allocation of namespaces to assignments .....	83
Figure 26: Integrated assignment authoring .....	83
Figure 27: Internal peer assessment on the difference page .....	84
Figure 28: Internal self assessment on the edit page.....	84
Figure 29: Rubric used for the group assessment .....	85
Figure 30: Group assessment feedback on the home page.....	85
Figure 31: Teacher's group assessment feedback.....	86
Figure 32: Internal peer feedback on the progress charts page.....	86
Figure 33: Integrated and rendered Wiki page.....	87

Figure 34: Contribution visualization on the group assessment page.....	87
Figure 35: Assignment home page.....	88
Figure 36: Progress charts page.....	89
Figure 37: Inline teacher feedback on the edit page .....	89
Figure 38: Individual contribution visualization on the teacher's view.....	90
Figure 39: Final grades page.....	91

## List of Tables

Table 1: Comparing mean scores of CSCL and non-CSCL writing groups (Hayes & Ge, 2008) .....	14
Table 2: Application of Assessment Criteria to Computer-Based Assessment (Higgins & Bligh, 2006) .....	28
Table 3: Student model used by ClassroomWiki (Khandaker & Soh, 2010) .....	36
Table 4: ACL operations of ScrewTurn Wiki for namespaces (ScrewTurn, 2012) .....	55
Table 5: Group-intern assessment and feedback activities.....	58
Table 6: Group-extern assessment and feedback activities .....	59
Table 7: Inline feedback colour codes.....	73
Table 8: Experimentations overview.....	92

# 1 Introduction

This introductory chapter should give a short overview of the main ideas and the motivation behind this project. Furthermore it outlines the structure of the remainder of this work.

## 1.1 Motivation and Objective

The emergence of the use of information and communication technology (ICT) in modern education settings has changed the way of how students work and learn. Especially the development of the Web from a static environment, where the content is mainly created by some experts, to the Web 2.0, the social Web, has raised the needs for new ways of *E-Learning* that support students in collaboratively creating content and knowledge. One of these types is called *Computer-Supported, Collaborative Learning (CSCL)* which is a research activity that focuses on approaches to support collaborative learning with computers and information technologies (Stahl, Koschmann, & Suthers, 2006). The most important implementation of such collaborative learning environments are collaborative writing tools such as Wikis or Blogs. However, these tools are usually designed for pure content creation and lack of important learning components like assessment activities. This is a big problem because assessment is much more than just grading a student's work and should be seen as an integral part of the teaching, learning and feedback processes (Buzzetto-More & Alade, 2006).

The aim of this master thesis is to identify a solution approach for combining Computer-Supported, Collaborative Learning with E-Assessment. Thus, an extensive literature research in the fields of Technology Enhanced Learning, CSCL and E-Assessment is done in order to understand the theoretical background. Based on this knowledge, related work is studied to identify problems, open issues and possible solutions. On the basis of these findings, a Wiki system is enhanced with assessment methods and other motivational tools in order to support students and teachers in collaborative writing activities. Moreover, this Wiki is also integrated in existing *Learning Management Systems* (LMS) to further enhance its functionalities. The next section outlines the structure of the remainder of this work.

## 1.2 Structure of the Work

This master thesis consists of two main parts: first, the theoretical part that aims at setting up the theoretical background for the second part that describes the practical developments of this work. The chapters 2, 3, 4 and 5 deal with the theoretical concepts and chapters 6, 7 and 8 concentrate on the practical part.

The first chapter of the theoretical part, chapter 2, acts as an introductory one and explains the most important ideas of Technology Enhanced Learning (TEL) and E-Learning. Chapter 3 looks at Computer-Supported, Collaborate Learning. This chapter has a focus on Computer-Supported, Collaborative Writing and Wiki systems as the most important implementation of it. Due to the fact that assessment is an integral part of every learning process, chapter 4 outlines the different types and strategies of it. Furthermore, this chapter explains how assessment methods can be enhanced by technology (E-Assessment).

Chapter 5 acts as a bridge between the theoretical and practical part and presents related work regarding the combination of CSCL with E-Assessment. This includes identified problems, issues and solution approaches that are used as an input for the enhanced Wiki that has been designed and developed as part of this thesis. The requirements and the design of this Wiki prototype are illustrated in chapter 6, whereas chapter 7 describes the development process of it. This is done for the standalone Wiki prototype and the integrated version of it. The last chapter of the practical part, chapter 8, explains the functionalities also from the user's point of view which includes the administrated experimentations too. Finally, chapters 9 and 10 reflect the whole work and show the lessons learned, a conclusion and possible approaches for future work.

## 2 Technology Enhanced Learning

Learning is one of the most important things in our everyday life and especially in the last decades the process of lifelong learning has evolved to a key factor for a successful career. But what does learning actually mean? According to Dror (2008) learning consists of two main elements, first the acquisition of information and second, the memory of information. This means that learning makes no sense if the learner only acquires the information and has forgotten it on the next day and so the learning process has to cover both elements to be successful. Based on this knowledge it is clear that a new learning technique can only support the process if it fits the human cognition. (Dror, 2008)

The aim of this chapter is to deal with the question if and how learning can be enhanced by technology. This research field is known as *Technology Enhanced Learning*, for short *TEL*. Thus, first an introduction to Technology Enhanced Learning is given which also includes a definition for this term and its used technologies. Furthermore, the applications of TEL are described with a focus on E-Learning as its most important form in today's learning settings.

### 2.1 Introduction to Technology Enhanced Learning

It is not easy to define the term Technology Enhanced Learning. Specht (2009) tries this with looking on the Wikipedia definition of the term Human Enhancement:

*“Human enhancement refers to any attempt to temporarily or permanently overcome the current limitations of the human body through natural or artificial means“*

In this sense, Technology Enhanced Learning refers to the enhancement of learning support via any kind of technology. However, this definition shows that the areas of TEL are very wide spread because they include every learning situation that is enhanced by technology, so also when a movie is shown in the classroom or when one student calls another student to discuss assignments. (Specht, 2009)

Technology Enhanced Learning is an extensive research activity that gets mainly influenced by three fields (Ebner, 2009):

- *Pedagogy* in terms of didactic design and the use of technology for learning and teaching

## Chapter 2: Technology Enhanced Learning

- *Computer science* in terms of technology, especially information and communication technologies
- *Human Computer Interaction* in terms of the interface between the learners and the technology

It is important that all of these three areas are mentioned when using TEL because otherwise an unbalance is the consequence. There already exist a lot of so called *Learning Management Systems (LMS)* such as Moodle<sup>1</sup> or Blackboard<sup>2</sup> to successfully realize TEL and its requirements. Usually these systems are also based on a specific standard for Technology Enhanced Learning such as the *IMS Question and Test Interoperability specification (QTI)*<sup>3</sup> or the *Sharable Content Object Reference Model (SCORM)*<sup>4</sup> to support interoperability between different systems. (Ebner, 2009)

## 2.2 E-Learning

Today almost everybody knows and uses the term *E-Learning*, which stands for *Electronic Learning*, although it is quite hard to give an exact definition for it, even amongst experts. Some people think that E-Learning is defined as sitting in front of a computer program to study the learning material instead of sitting in the classroom to learn via classical face-to-face methods. In fact it is far more than only this; actually it is any form of learning where technology is used beyond the pedagogical aspect. This definition is very similar to the one for TEL and so it is also often defined for situations where the learning happens via computers or the internet to make a distinction. (Nichols, 2008)

The core idea of E-Learning has even evolved at about the same time as the first forms of distance learning were used although there can be found some differences between those types. In 1840, in England, courses were offered where the learning material was distributed by mail. It is clear that this type of distance learning has nothing to do with modern technologies, which is a key factor for E-Learning, but e-mails or computers have only made this approach better without changing the key principle of it. (Aranda, 2005)

Another very important step in the development of today's E-Learning systems was the change from systems that were based on Computer-Based Learning and Web-Based Learning (see section 2.3) to systems that used the idea of the Web 2.0, the social Web.

---

<sup>1</sup> <http://www.moodle.org/>; visited on 05<sup>th</sup> of December 2011

<sup>2</sup> <http://www.blackboard.com/>; visited on 05<sup>th</sup> of December 2011

<sup>3</sup> <http://www.imsglobal.org/question/>; visited on 05<sup>th</sup> December 2011

<sup>4</sup> <http://scorm.com/>; visited on 05<sup>th</sup> of December 2011

## Chapter 2: Technology Enhanced Learning

The first ones are assumed to offer knowledge, which was created by teachers, to learners and the second ones are based on the concept to give the learners the opportunity to create knowledge by themselves. This is also known as the social learning idea of E-Learning 2.0 which is often implemented via Computer-Supported, Collaborative Learning systems that are further explained in chapter 3. (Brown & Adler, 2008)

### 2.3 Approaches of Technology Enhanced Learning

As mentioned before, Technology Enhanced Learning can be used for different purposes and so there are a lot of approaches to realize it. The following listing should give an overview of the most important concepts in this context:

- **Computer-Based Training/Learning**

*Computer-Based Training or Learning (CBT or CBL)* is a form of TEL where learners follow the steps of a learning program. Such programs can be used for example to learn foreign languages or computer applications. (E-Teaching, 2010)

- **Web-Based Training/Learning**

*Web-Based Training or Learning (WBT or WBL)* is used in a similar way as CBT/CBL, but here the learning material is distributed by the World Wide Web (WWW). This has the great advantage that communication technologies between learners can be used and that teachers can easily update the material online. (Helic, Maurer, & Scerbakov, 2000).

- **Computer-Supported, Collaborative Learning**

*Computer-Supported, Collaborative Learning (CSCL)* describes approaches how collaborative learning and writing can be realized with the help of computers (Stahl, Koschmann, & Suthers, 2006). CSCL is described in details in chapter 3.

- **E-Assessment**

The term *E-Assessment* describes methods and ways to support assessment with modern information and communication technology, for short ICT (Brahm & Seufert, 2007). Chapter 4 deals with E-Assessment in detail.

- **Blended Learning**

*Blended Learning* is the combination of the classical learning and teaching method in the class room with E-Learning sessions that are provided online. This form of TEL is a good agreement to get the advantages of both methods. (Ebner & Schoen, 2011)

## 2.4 Summary

It is not easy to define the term Technology Enhanced Learning (TEL), but according to Specht (2009), TEL refers to the enhancement of learning support via any kind of technology. To successfully use TEL it is important to mention all three areas of it: Pedagogy, Computer Science and Human Computer Interaction. Otherwise an unbalance is the consequence. TEL is often realized with so called Learning Management System (LMS) such as Moodle or Blackboard.

E-Learning is the most important form of TEL and it is defined as situations where the learning happens via computers or the internet. The principles of E-Learning go back to the year 1840 when the first distance learning courses were offered in England. Although no computers were used for these courses, modern learning methods like E-Learning 2.0 are still based on the same main idea. Other approaches for TEL and E-Learning are: Computer-Based Training/Learning (CBT/CBL), Web-Based Training/Learning (WBT/WBL), Computer-Supported, Collaborative Learning (CSCL), E-Assessment and Blended Learning. The next chapter gives an overview of Computer-Supported, Collaborative Learning with a focus on Computer-Supported, Collaborative Writing and especially Wiki systems.

## 3 Computer-Supported, Collaborative Learning

The aim of the last chapter was to give a general idea of Technology Enhanced Learning and especially E-Learning and introduced their most important types of it. One of these types is *Computer-Supported, Collaborative Learning (CSCL)* which is a research activity that focuses on approaches to support collaborative learning with computers and information technologies. In contrast to supply-side E-Learning systems, self-regulated CSCL environments are a young research topic and a lot of studies are based on results of the Computer-Supported, Collaborative Working (CSCW) research, which is a CSCL approach that has its focus on economic group work instead of learning. (Stahl, Koschmann, & Suthers, 2006)

The aim of this chapter is to give a detailed overview of CSCL and especially Computer-Supported, Collaborative Writing. After listing the basic terms and definitions of CSCL, a theoretical and practical outline of collaborative software (groupware) and learning tools is given. These explanations lead to the main section of this chapter, Computer-Supported, Collaborative Writing. Here the principles of this research area are described with a focus on Wiki systems.

### 3.1 Introduction to Computer-Supported, Collaborative Learning

This section gives an introductory overview of the basic terms of Computer-Supported, Collaborative Learning to form a better understanding for collaborative software and learning tools and especially collaborative writing which is the main topic of this chapter. It gives a definition of the term CSCL, describes its history and the motivation behind it and figures out the aspects and principles of it.

#### 3.1.1 Definition and Aspects of Computer-Supported, Collaborative Learning

Computer-Supported, Collaborative Learning describes approaches how collaborative learning can be realized with the help of computers. This means that learning groups should create and share knowledge with the primary use of communication features that are offered by computer technologies such as chats or forums. Another perspective of CSCL is

## Chapter 3: Computer-Supported, Collaborative Learning

the challenge of combining learning with modern technologies and how this can be solved. (Stahl, Koschmann, & Suthers, 2006)

To realize CSCL, learning groups need a computer-supported environment that offers at least the following five components (Nohr, 2005):

- Coordination of the learn groups using some sort of workflow tool
- Access to the shared learning material and its annotations
- Possibilities to work in a collaborative way on new material as part of home works or presentations
- Synchronous or asynchronous communication of their knowledge
- Group management that also supports an role model

These computer-supported learning environments can either be implemented as distributed systems, so that the individuals are connected from different places and maybe also at different time, or as local applications in so called CSCL laboratories, so that the individuals learn physically together (Nohr, 2005).

### 3.1.2 Motivation and Principles for Supporting Collaborative Learning with Modern Technologies

Today's research activities in the fields of learning and E-Education look for new ways of more learner-centered methods in order to promote social creativity. This should give students the opportunity to increase their collaborative working skills and to exchange their knowledge among each other. Especially computer-supported environments, which also have a long history in collaborative working, seem to be perfect to support these learning processes because they offer multiple ways for easily bringing together different points of view and resources. The great benefit here is that the social debates and discussions, feedbacks and knowledge construction processes, which are built around these brought-in resources, can create new ideas, insights and also new resources. (Ma, 2008)

To make use of CSCL, the supporting computer environments must be designed in a way to initiate social interactions that are necessary for transferring knowledge. Dillenbourg and Fischer (2007) mentioned some ideas and principles to do this:

- Placing students into situations where they need to interact to build a shared understanding
- Using a representation of the tasks that shape the language used by students
- Structuring the processes by guidelines, so called collaborative scripts

- Capturing and analyzing interactions and showing these to other group members
- Providing scaffolds to students to better understand the process

When applying these principles, social interactions can be designed in small collaborative learning environments like we face them in learn groups or school classes. For larger and often virtual communities the research activities are still in their early ages because a much deeper understanding of knowledge building and exchange processes are needed. (Dillenbourg & Fischer, 2007)

### **3.1.3 History of Computer-Supported, Collaborative Learning**

The principles of CSCL goes back very far but one of the early bigger projects was called ENFI (*Electronic Networks for Interaction*) and started in the year 1985 at the Gallaudet University in Washington, D.C. The Gallaudet University is a university for deaf and hard-of-hearing people and so the ENFI project focused on giving the students the possibility to use written English, in the same way as hearing people use spoken English, for their learning purposes. This means that they can chat to discuss their academic texts and to generate ideas, which can also be seen as the first step in the development of Computer-Supported Collaborative Writing because this can also lead to great benefits for hearing students. (Peyton, 1990)

The first time the term Computer-Supported Collaborative Learning was officially used was in the year 1989 at a workshop in Maratea, Italy that was financed by the NATO. The first complete CSCL based conference was held in 1995 at the University of Indiana which was continued every two years in cities such as Toronto, Stanford, Bergen and Taiwan. At this time also CSCL based books by Springer and Journals like the *International Journal of Computer-Supported Collaborative Learning* were released that discussed ways how students could be brought together with the help of computers to generate knowledge together instead of getting knowledge by a teacher or by materials that were created by a teacher. (Stahl, Koschmann, & Suthers, 2006)

## **3.2 Collaborative Learning Tools**

Collaborative software (Groupware) and collaborative learning tools are the technology to realize the research areas of Computer-Supported, Collaborative Learning and Working. This means that it gives distributed users the possibility to collaborate through some kind of infrastructure, to generate products together and in the best case also new individual knowledge. (Rama & Bishop, 2006)

The remainder of this section is divided into two parts. First the learning activities (*E-tivities*) that have to be fulfilled by a collaborative learning tool are discussed to get a theoretical background of the workflow and the requirements of such a system. The second part gives an overview of collaborative software and learning tools.

### 3.2.1 E-tivities

Salmon (2002) argues that the future of collaborative learning tools lies in online activities which are referred to as so called E-tivities. In her book *E-tivities: The key to active online learning* she describes a conceptual framework to show how these online activities can be created and how they affect the design and the technical requirements of collaborative tools. Furthermore, this framework, which is known as the Five-Step Model (shown in Figure 1), takes a look on how the teacher, or online instructor, can support the activities to increase the participation and interaction of the students.

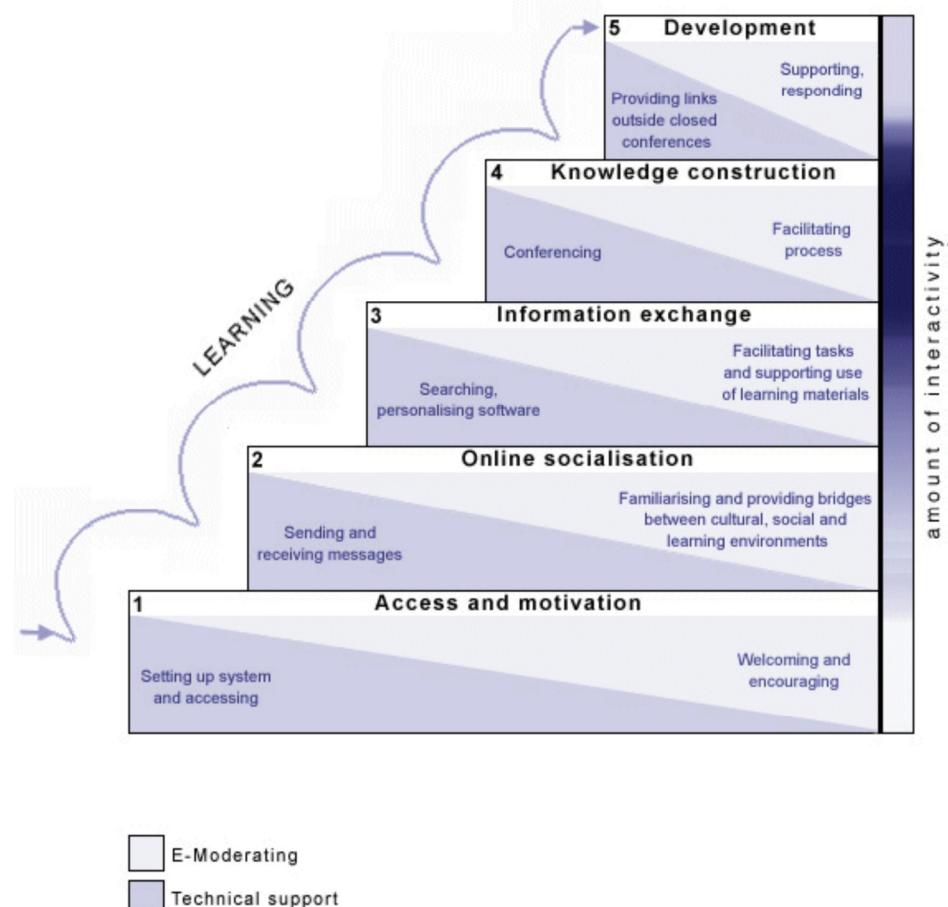


Figure 1: Five-Step Model to create E-tivities (Salmon, 2002)

## Chapter 3: Computer-Supported, Collaborative Learning

The online learning process for the students starts with accessing the system to try out its possibilities. Here it is important that relevant instructions are provided to motivate the learners to further use the tool, because they can be easily embarrassed if they do not know how to use the system. Furthermore, the online instructor should be available in case of problems that often occur in the first steps. After this step the online socialization begins where the students get to know their group members, and so some kind of communication medium, such as a discussion forum, a chat or a messaging system, has to be offered. In case of obstacles caused by different languages or other social problems, the instructor has to help to start the communication. In step 3 the students receive their tasks and learning materials and begin to share their knowledge and findings among each other. These knowledge-exchange processes lead to the knowledge-construction step where the students use the discussions with their peers to create new knowledge. During this step, or latest during the last step, the students use the working area of the system (in case of a Wiki this would be a Wiki page, where they can edit their content) to solve the problems of their tasks in collaboration with their peers. (Salmon, 2002)

### 3.2.2 Types of Collaborative Software and Learning Tools

To identify the different types and forms of collaborative software it is important to understand the two dimensions of Computer-Supported, Collaborative Learning and Working which are time and space. This means that it is not only possible that people work or learn together in a distributed manner or at the same place, but also at the same time (synchronous) or at a different time (asynchronous). Rama and Bishop (2006) visualized this in the quadrants that are shown in Figure 2.

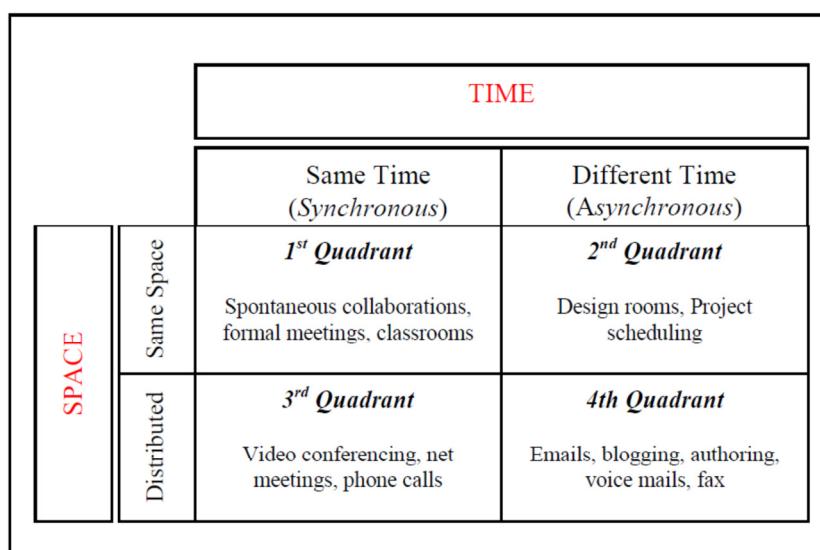


Figure 2: Quadrants to show the dimensions of CSCL and CSCW (Rama & Bishop, 2006)

## Chapter 3: Computer-Supported, Collaborative Learning

To understand these quadrants, Rama and Bishop (2006) used an example of two collaborators that work on the same project. If they are together in the same office at the same time they can use normal conversations to share their knowledge and so they fall into the quadrant 1. In cases where they two work at different times, but share the same space, whiteboards could be used for persist their ideas over time (quadrant 2). However, in distributed projects it is unusual that the project members share the same place, so they would fall into the quadrants 3 and 4. For synchronous communication normally phone calls or video conferences, and for asynchronous communications e-mails or forms of collaborative writing, are used.

The focus which is of interest in the context of this research falls into quadrant 4 and so the following listing should give some examples of the possible communication mediums and tools that are part of it:

- Communication tools such as e-mail, fax, voice-mail or internet forums
- Management tools such as electronic calendars, workflow systems or social software
- Web-publishing methods using Web pages or services like GoogleDocs
- Revision control systems for software development such as CVS, SVN or Mercurial
- Groupware systems such as Novell iFolder, Microsoft SharePoint or Apple SubEthaEdit
- Collaborative writing systems such as Wikis or Blogs

This research focuses on Computer-Supported, Collaborative Writing and especially Wiki systems which will be described in the next section.

### 3.3 Computer-Supported, Collaborative Writing

Computer-Supported, Collaborative Writing is the process of supporting joint writing tasks with the help of collaborative writing tools such as Wikis or Blogs in order to shorten the writing time and to improve the final product (Noel & Robert, 2004).

This section first describes the processes and complexities of collaborative writing and how they can be positively affected by CSCL tools. Furthermore, it gives an overview about one of its most important technologies, Wiki systems.

### 3.3.1 Process and Complexity of Collaborative Writing

Collaborative writing is a very actual topic because the majority of all written work is created in collaboration and so it is clear that the nature of the collaboration varies from activity to activity. Sometimes there is only a single author who is supported by others for reviewing and proofreading and other times there are multiple authors that work together tightly coupled. The form with the multiple authors has the big advantage that a lot of different knowledge and ideas is brought together, but also has the disadvantage that much organizational effort is needed. (Tammaro, Mosier, Goodwin, & Spitz, 1997)

To describe the complexity of collaborative writing Kraut, Galegher, Fish and Chalfonte (1992) define three types of complexity of creating joint documents:

- Social complexity
- Intellectual complexity
- Procedural complexity

*Social complexity* contains the social finesse in defining goals together and dividing tasks among the group members, whereas *intellectual complexity* includes the processes of getting a common understanding of the facts and details the document will be based on. Finally the group members are also faced with *procedural complexity* because they have to manage it to handle and accept changes from others and to also respect their comments. (Kraut, Galegher, Fish, & Chalfonte, 1992)

### 3.3.2 Effects of CSCL on Collaborative Writing

Although the collaborative writing process has been used in years before computers were part of our everyday life with the help of paper and pencil, recent studies show that a Computer-Supported, Collaborative Learning environment can have positive effects on students' writing performance. This is because the students are supported by tools in their collaborative writing not only by fancy writing and editing possibilities, but also by complete histories of all revisions to get a better overview of the changes they have done.

One user study has been done by Hayes and Ge (2008) at an urban elementary school in the southwest of the United States with a tool called Knowledge Forum<sup>5</sup>. For the study 34 fifth grade students, in age from 10 to 13, were divided into two groups and were asked to complete a joint writing assignment, one group with and the other group without the help of CSCL. For the grading a scoring schema (rubric) with five sections, where each

---

<sup>5</sup> <http://www.knowledgeforum.com>; visited on 07<sup>th</sup> of October 2011

## Chapter 3: Computer-Supported, Collaborative Learning

was worth 3 points, was used and so a maximum of 15 points could have been reached. The scoring schema with the mean and standard derivation values can be seen in Table 1.

	<b>Group</b>	<b>Mean</b>	<b>SD</b>	<b>Sample Size</b>
Idea Development	CSCL	3.28	.75	18
	Non-CSCL	2.44	1.09	16
Organization	CSCL	2.72	.75	18
	Non-CSCL	2.19	.83	16
Word Choice	CSCL	2.56	.71	18
	Non-CSCL	1.88	.96	16
Sentence	CSCL	2.5	.86	18
	Non-CSCL	1.81	.75	16
Grammar	CSCL	2.39	.92	18
	Non-CSCL	1.94	.77	16
<b>Composite</b>	CSCL	13.44	3.55	18
	Non-CSCL	10.25	4.09	16

Table 1: Comparing mean scores of CSCL and non-CSCL writing groups (Hayes & Ge, 2008)

Findings suggest that the students in the CSCL had higher mean score values in all sections of the grading schema than the students that did not have any computer-support. This means that CSCL usually improves the collaborative writing skills and enhances the writing performance of students. (Hayes & Ge, 2008)

According to Kimmerle, Moskaliuk and Cress (2009) new developments of the Web also lead to new developments in learning and teaching. Especially the big step to Web 2.0, with the developments of tools such as Wikis, Blogs, Podcasts and virtual online worlds, had a great impact on collaborative learning and writing. The next section focuses on an overview about Wiki systems because also the practical part of this master thesis is based on such a system.

### 3.4 Wiki Systems

Nowadays almost every internet user knows the Web encyclopedia Wikipedia<sup>6</sup> and uses it nearly every day for looking up things but only a few know that it is built on a Wiki system and so opens the possibility for everyone to create and edit content. To get a better understanding of such Wiki systems, this section explains the characteristics of them and how they can be used in education.

---

<sup>6</sup> <http://www.wikipedia.org>; visited on 07<sup>th</sup> of October 2011

### 3.4.1 Definition and Characteristics of Wiki Systems

In their well-known book “*The Wiki Way – Quick Collaboration on the Web*” Leuf and Cunningham (2001) define a Wiki system with the following words:

*“A Wiki is a freely expandable collection of interlinked Web “pages”, a hypertext system for storing and modifying information – a database, where each page is easily editable by any user with a forms-capable Web browser client”*

This means that the pages in a Wiki system act like “real” Web pages, so they can be reached by the URL concept (*Unified Resource Locator*) and they can have incoming and outgoing hyperlinks, but the content of the pages are saved in the database of the Wiki system and not in HTML files (*Hypertext Markup Language*). This concept makes it possible that the Wiki pages can be easily modified and also rolled-back to previous versions by anyone who has the necessary permissions for it. (Leuf & Cunningham, 2001)

Ward Cunningham also was the inventor of the Wiki idea and developed the first Wiki system called WikiWikiWeb<sup>7</sup>. The term “Wiki” is inspired by the Hawaiian word *Wikiwiki*, which stands for “fast” or “quick” and denotes that creating content with Wikis is a very easy process. To sum this up Leuf and Cunningham (2001) define the following characteristics of Wiki systems:

- Every user can create or edit Wiki pages, only using a Web browser and a very simple Wiki markup language (no HTML is needed)
- Topic associations can be made very easy with the help of nearly automatically created links (only the Wiki page name is needed)
- A Wiki page is not intended to be a static view-only page because it invites users to be part of an iterative content- and knowledge-creation process.

### 3.4.2 Wikis in Education

Learning with the help of Wikis is a form of self-regulated learning that takes place in groups of collaboratively working people. This means that the learning groups define their goals and learning/writing strategies in a self-dependent way to create new knowledge. This new knowledge is created via Wikis as interplay of a social system and a cognitive system. Here the Wiki and the learning community refer to the social system and the cognitive system is defined by individual processes such as thinking, remembering and reasoning about things. (Kimmerle, Moskaliuk, & Cress, 2009)

---

<sup>7</sup> <http://c2.com/cgi/wiki?WikiWikiWeb>; visited on 7<sup>th</sup> of October 2011

Wikis are used in education since 1997 when researchers at the Georgia Institute of Technology implemented a variation of Ward Cunningham's WikiWikiWeb system, called *CoWeb*. It was used by teachers to provide easily editable learning materials or as a construction kit for students to handle collaborative writing assignments. One use case was also to create a collection of computer programming design patterns (Forte & Bruckman, 2007). The next subsections describe possible usage scenarios for Wiki systems in education and examples of educational Wiki systems are given. A more detailed explanation of how Wikis can be used together with flexible E-Assessment methods can be found in chapter 5.

### 3.4.2.1 Usage Scenarios for Wiki Systems in Education

The use of Wikis in education has increased a lot in the past few years because of the open concept of it. Generally there are two different ways of using educational Wikis: as a tool to support collaborative writing projects for students and as repositories for creating learning materials. In the second scenario it is also possible that teachers get supported by students to create and share learning contents or that multiple teachers work together to do this. (Biuk-Aghai, Kelen, & Venkatesan, 2008)

This also refers to the idea of the Web 2.0, the participatory Web, where the user not only consumes content but also creates it. This means in terms of educational Wiki systems that also the students can create learning material which can be consumed and edited by other students. Another example of such learning repositories is the PBS Wiki<sup>8</sup> that was fully created by students of the University of Technology in Graz to collaboratively collect questions and answers of old exams to help colleagues to prepare for new exams.

Duffy and Bruns (2006) identified the following educational usage scenarios for Wikis systems:

- Students can develop research projects with a build-in log using the Wiki's revision history. This gives them a good overview of their working process.
- Wikis can be used for brainstorming and other creative activities where topics get linked together to produce visualizations of concepts.
- In distance education settings, exercises can be offered to the students as Wiki pages to give them the opportunity to directly edit them (e.g. for language learning).

---

<sup>8</sup> <http://pbs.htu.tugraz.at/wiki/index.php/>; visited on 23<sup>rd</sup> of January 2012

- Teachers can use Wikis as easy searchable, navigable and editable knowledge repositories for their learning materials. Furthermore, also file uploads are usually supported in Wikis.
- Wikis are tools for supporting collaborative writing projects. When using programs and technologies such as Microsoft Word, LaTeX or Open-Office for writing projects, students often sharing the documents via e-mail or versioning systems like SVN, which produces extra workload for the students. With Wikis only a simple Web browser is needed for working and writing together.
- The open concept of the Wiki makes it also very useful for course reviews and evaluations. The Brown University in Providence, Rhode Island already uses a special Wiki for this purpose.

Beside this powerful usage scenarios and benefits, Wikis also have educational challenges, which have to be mentioned. First it is important that the Wiki offers an appropriate permission system on the level of pages because usually it is necessary that there are special pages only available for teachers or that students should not see the pages of their peers. Furthermore, the Wiki should use some kind of page-lock mechanism when a page gets edited by two or more students. Otherwise later changes during the same edit-session would overwrite the earlier ones. (Duffy & Bruns, 2006)

### 3.4.2.2 Comparison of Educational Wiki Systems

Since Ward Cunningham's WikiWikiWeb system, which was written in Perl, a lot of other Wiki systems have been developed. These systems differ in the programming technologies they are implemented with, in the database systems they use as a persistent data backend and in the decision if they are used in distributed systems, like the Web, or only locally. There also exist so called Wiki farms which offer Web interfaces that let users create Wiki systems with only a few clicks. These farms also host the Wikis for their users and so no Web server installation and configuration is needed. Challborn and Reimann (2005) have compared different Wiki systems and Wiki farms and have discussed their educational potential:

- **MediaWiki**

MediaWiki<sup>9</sup> is an open-source Wiki written in PHP and uses a MySQL backend database. It is one of the best-known Wiki systems because it is also used by the Wikipedia Web encyclopaedia. It offers a lot of configurations from page-level

---

<sup>9</sup> <http://www.mediawiki.org/>; visited on 04<sup>th</sup> of January 2012

permissions to namespaces for pages, which makes it very flexible but also quite complicated for teachers. For educational uses its discussion forums, revision histories and summaries and plugins for citations or WYSIWYG editors can be very helpful. Other well-known stand-alone and open-source Wikis are the Microsoft .NET based ScrewTurn Wiki<sup>10</sup> and the Perl based TWiki<sup>11</sup>, that even supports reports.

- **EditMe**

EditMe<sup>12</sup> is a commercial Wiki farm for creating and hosting Wikis. It is very easy to use and so can also be used by teachers with no technical background, but lacks detailed configuration possibilities that are offered by systems like MediaWiki. For educational purposes it offers a commenting system, ratings for page assessments, WYSIWYG editors, TODO-lists and a merging mechanism if two students concurrently edit the same page. Examples for other well-known Wiki farms are Wikispaces and WetPaint.

- **Swiki**

Swiki<sup>13</sup> is an open-source Wiki system written in the Smalltalk based programming language Squeak. It is an implementation of Ward Cunningham's original WikiWikiWeb system and uses a XML file based data backend. For education Swiki offers similar Wiki functionalities as MediaWiki, but does not support file attachments, which makes it useless for most teachers. On the other hand it comes with its own Web server and runs in a virtual machine, so no complicated installation and configuration is needed.

- **Socialtext**

Socialtext<sup>14</sup> is a commercial enterprise social software tool for collaborative working. It tries to manage collaborative projects and information in a social network environment with tools such as activity streams, Wikis and blogs. These social awareness features makes it ideal for educational purposes, but since it is a commercial and expensive tool it is not an option for typical educational users.

Based on the listing above, standalone Wikis, like MediaWiki, are the best choice for educators with a technical background. Since these Wikis are free and open-source, it is possible to enhance them to fulfil special educational needs. However, users that do not have the

---

<sup>10</sup> <http://www.screwtturn.eu/>; visited on 04<sup>th</sup> of January 2012

<sup>11</sup> <http://twiki.org/>; visited on 04<sup>th</sup> of January 2012

<sup>12</sup> <http://www.editme.com/>; visited on 04<sup>th</sup> of January 2012

<sup>13</sup> <http://wiki.squeak.org/swiki/>; visited on 04<sup>th</sup> of January 2012

<sup>14</sup> <http://www.socialtext.com/>; visited on 04<sup>th</sup> of January 2012

knowledge for setting up a Wiki system should stick to a Wiki farm like EditMe. (Challborn & Reimann, 2005)

### 3.5 Summary

Computer-Supported, Collaborative Learning (CSCL) is a research area that focuses on approaches to support collaborative learning with the help of computers and information systems. In CSCL these information systems are called computer-supported learning environments. These environments can either be implemented as distributed systems, where the individuals learn from different places and maybe also at different time, or as local applications, where the individuals can learn physically together. To get the benefits out of CSCL, it is important that the students are placed in situations where they have to build a shared understanding of the topics and material among each other. This makes it easy to bring together different point of views in order to create new insights, ideas and knowledge.

CSCL can be realized with collaborative software and collaborative learning tools. These tools have to be built around online activities, so called E-tivities. Salmon (2002) described these E-tivities in her Five-Step Model (*Access and motivation, Online socialisation, Information exchange, Knowledge construction, Development*). Collaborative learning tools can be divided into four groups, or quadrants, depending on the two dimensions of CSCL, time and space. Usually collaborative writing tools such as blogs or Wikis fall into quadrant 4, what means that they are used from distributed places and at different time (asynchronous).

Collaborative writing is the process of supporting joint writing projects with the help of such collaborative writing tools in order to tackle the complexities of it (*social, intellectual, procedural*). Wiki systems are one way to realize collaborative writing environments and also the practical part of this thesis is based on a Wiki system. Cunningham and Leuf (2001) define a Wiki with the following words: “*A Wiki is a freely expandable collection of interlinked Web “pages”, a hypertext system for storing and modifying information – a database, where each page is easily editable by any user with a forms-capable Web browser client*”. This means that the content of a Wiki page is saved in a database and not in a HTML file and so anyone is able to edit it and is also able to see and roll-back all previous versions of it.

Educational Wikis are used for self-regulated learning where the student groups define their writing goals in a self-dependent way to create new collaborative knowledge. This type of learning happens as interplay of a social system (the Wiki itself) and a cognitive system (processes such as thinking, reasoning and remembering). Wikis are mainly used in

## Chapter 3: Computer-Supported, Collaborative Learning

educational settings as knowledge repositories for learning material, for collaborative writing projects, as a tool for brainstorming or for course evaluations. Challborn and Reimann (2005) have compared different Wiki systems and Wiki farms and have analyzed their usage for educational settings. According to their findings, standalone Wikis, like MediaWiki, are the best choice for educators with a technical background. Users that do not have the knowledge for setting up a Wiki system should stick to a Wiki farm like EditMe. The next chapter will give an overview of E-Assessment in order to finish the theoretical background that is needed to understand and to feed into the practical part of this thesis.

## 4 E-Assessment

Assessment is defined as an on-going process that consists of documenting, measuring, analysing, reflecting and discussing data and artefacts that were gathered about a learning objective. According to this definition assessment is much more than just grading a student's work and should be seen as an integral part of the teaching, learning and feedback processes. When correctly designed assessment can not only be used to improve the learning of students and to identify their strengths and weaknesses but also to review and improve the effectiveness of teachers and curriculum programs. (Buzzetto-More & Alade, 2006)

The aim of this chapter is to explain how assessment can be realized and supported by modern information and communication technologies. Thus, first an introduction to the term E-Assessment, that covers also its history and the motivation behind it, is given. After this the different types and strategies of E-Assessment are described to define the necessary terms that are needed to understand the third section, which focuses on the combination of assessment and computer science. The fourth and last section of this chapter gives a short introduction to assessment rubrics. This means that this section takes a further look on the different forms of E-Assessment and on its advantages over traditional paper-based assessment.

### 4.1 Introduction to E-Assessment

The term E-Assessment describes methods and ways to support assessment with modern information systems. Often E-Assessment gets confused with the concept of E-Portfolios and also some people think that they are the same, but E-Portfolios have a different purpose. Whereas the first one is mainly used for grading and feedback provision, the second one is a collection of data and documents, that describe the learning process of a student, and is used for documentation and reflection (Brahm & Seufert, 2007). The remainder of this section gives a more detailed definition of the term E-Assessment and describes its history. Furthermore, the motivations and rationales for its use are given.

#### 4.1.1 Definition and History of E-Assessment

E-Assessment can be defined as all the methods that are used to define, evaluate, rate, document and feedback learning processes before, during and after the teaching period. The important point here is that these methods are based on electronic information and communication technologies to support students and teachers to reach their learning targets.

## Chapter 4: E-Assessment

This broad definition of E-Assessment shows that it is a complex research area and so also different types and subtypes of it, such as Computer-Based assessment (CBA), Computer Assisted-Assessment (CAA) and online assessment, have evolved (Brahm & Seufert, 2007). They will be further explained in subsection 4.2.

The history of the use of computers to support assessment tasks goes back for decades but one of the earliest projects was called PLATO (*Programmed Logic for Automated Teaching Operations*) and was developed in the early 1960's at the University of Illinois. Another important research area in the early days of E-Assessment was the automatic assessment of students' programming assignments. One of these assessment systems was called the *Automatic Grader* which was not only used to automatically compile and test students' programs but also to help students to increase their programming skills and to support teachers in supervising large programming classes.

In the 1980's the interest in using computers for instruction increased a lot and so also formative assessment methods were used to evaluate the drafts of the instructional material before it was used in class. The next milestone in the history of E-Assessment was the rise of the *World Wide Web*, for short WWW, in the 1990's. With this new technology the first Web based assessment systems such as Blackboard<sup>15</sup> or ASSYST were developed and published (AL-Smadi & Guetl, 2008). Today's research in the field of E-Assessment goes towards fully automatic assessment systems for open-ended test items as described by Guetl (2008). Another example for this is the automatic creation of questions based on learning material (Guetl, Lankmayr, Weinhofer, & Hofler, 2010).

### 4.1.2 Motivations for the Use of E-Assessment

As mentioned above, E-Assessment is the use of assessment methods with the help of modern information and communication technologies. According to AL-Smadi and Guetl (2008) there are two main motivations for supporting assessment processes with computers, practical and pedagogic ones:

- **Practical rationales**

The number of students continuously increases every year and so also the assessment workload of teachers and tutors increases with the same factor. To tackle this problem an E-Assessment system can be very helpful, especially when they support automatic ways to provide continuous feedback for students.

---

<sup>15</sup> <http://www.blackboard.com/>; visited on 2<sup>nd</sup> of November 2011

- **Pedagogic rationales**

The assessment during learning, also known as formative assessment is a very important aspect of every learning process. When using classical methods it is hard for teachers and tutors to provide a fair, consistent, reliable and efficient assessment. Computer based systems can help to fulfil these qualitative needs when they are designed correctly.

Based on these rationales and according to Buzzetto-More and Alade (2006) E-Assessment can be used for various assessment processes such as student tracking, diagnostic analysis, pre and post testing, artefact collection, data aggregation and the use of assessment rubrics. Especially assessment rubrics, which are assessment tools that can be used as consistent scoring schemas, have great benefits when combined with digital formats. With the help of a database system, they can be aggregated and used by teachers in a flexible way according to their needs. Subsection 4.4 describes assessment rubrics in detail. Furthermore, the authors mention some advantages of computer-supported assessment over traditional assessment in the following areas: reliability, scoring, data management, cost, usage and flexibility.

## 4.2 How to Differentiate Assessment Forms?

Over the years a lot of different assessment forms have been developed. Usually they can be differentiated based on their strategy and type and so this section gives an overview about this. Furthermore, this section should act as some sort of index to give general definitions for the subsequent chapters and sections.

### 4.2.1 Strategies of Assessment

Assessment is a complex task and is used for different types of learning processes. To cover all purposes three main assessment strategies have evolved that were visualized by Crisp (2009) in Figure 3. It shows that the three assessment strategies build a cycle around the learning process, the so called assessment cycle. This cycle should fulfill two main purposes, the assessment for learning and the assessment of learning. The first one, assessment for learning, is used by students to improve their learning skills based on the given continuous feedback and by teachers to improve their teaching based on the assessment data. The second one, assessment of learning, is usually used by teachers for giving marks, so it covers the classical view on assessment (Earl, 2003). The following listing describes the three assessment strategies of Figure 3:

- **Formative assessment (assessment during learning)**

The aim of formative assessment is to assist the learning process and so it should occur during it. The most important outcome of this assessment strategy is formative feedback.

This formative feedback should fulfill the following principles: assist self assessment; bring forward peer and teacher communications; give high quality information to students and the opportunity to improve based on this information; provide information to teachers to improve their teaching. (Higgins & Bligh, 2006)

- **Summative assessment (assessment after learning)**

Opposed to formative assessment, the primary aim of summative assessment is to get an indicator of the learning process and so it is usually done with a test at the end of it. There exist internal and external uses of this assessment strategy. Internal uses include school tracking of the progress of the students and informing students, students' next teachers and parents of what has been accomplished. External uses include the selection for employment or further education and for examining the performance of schools and teachers. (Harlen, 2005)

- **Diagnostic assessment (assessment before learning)**

Diagnostic assessment is undertaken before the learning process has started and is typically used for skill gap analyses. This means that it could be used by students to determine if their prior knowledge is good enough to attempt a specific course or by teachers to determine the knowledge level of a class to know where to start with their learning material. (Crisp, 2007)

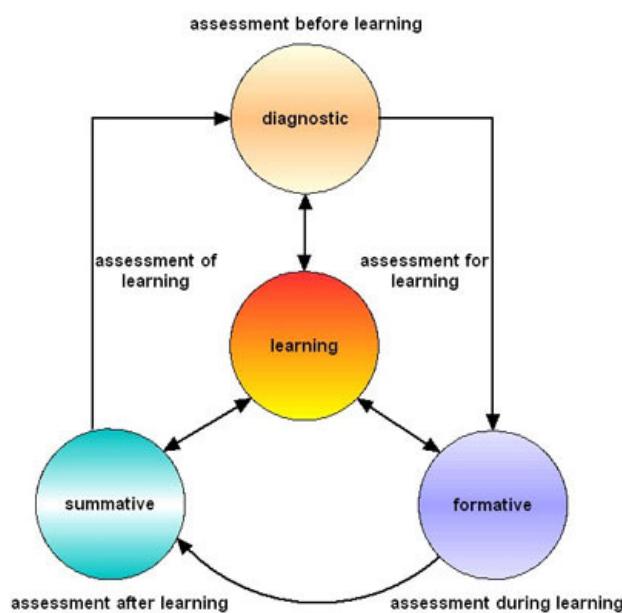


Figure 3: Strategies of assessment (Crisp, 2009)

## 4.2.2 Types of Assessment

The second way to differentiate assessment forms is based on their usage type. Usually the types shown in the following listing can be used either with a summative or formative purpose:

- **Teacher/Tutor assessment**

This is the classical type of assessment where the marks are only given by the teachers or tutors. It can be used for marking and also for course improvements. (Roberts, 2006)

- **Peer assessment**

In peer assessment students provide other students, their peers, with grading and feedback. This type of assessment is used to increase the quality of learning because the students have to carefully analyze the work of their peers to give a consistent and fair feedback. Furthermore, peer assessment decreases the workload of teachers and tutors. (Bostock, 2000)

- **Self assessment**

In contrast to peer assessment, self assessment is used to give students the possibility to reflect and to suggest grades for their own work. This gives the great benefit to the students that they have a higher engagement with their own learning process, which let them determine their weak points to train them. (Roberts, 2006)

- **Group assessment**

Group assessment can be used and defined in multiple ways. It can either be used to assess the work of whole groups by teachers or other students or to let the students assess the work of their group members among each other. The second usage is very close to the one of peer assessment and shows that peer assessment can also be helpful for assignments that are based on group work. (Roberts, 2006)

- **Collaborative assessment**

Collaborative assessment, or also called co-assessment and cooperative assessment, is a way of providing the possibility to students to assess themselves while the control of the assessment process and the final mark stay at the side of the teacher. This type of assessment is often used for summative purposes. (Dochy, Segers, & Sluijsmans, 1999)

- **Automatic assessment**

In automatic assessment environments the assessment is automatically done by a computer system. This can be easily implemented for question types such as single-choice or true-false, but needs a lot of effort for questions that await short free-text answers. (Guetl, 2008)

## 4.3 Assessment and Computer Science

Due to the increasing number of students each year and the unchanged need for valuable and consistent feedback, new forms of technology enhanced assessment have evolved. It is sometimes portrayed that the use of computers for assessment will automatically improve the student outcomes and assessment processes, but actually E-Assessment methods have to be as carefully designed as traditional assessment methods (Crisp, 2007). This section should explain how traditional assessment methods can be enriched with computer science. Thus, the different types of E-Assessment are described and also the benefits of computer-supported assessment over paper-based assessment are given. It should be mentioned that this section only acts as an overview and that more specific methods of how modern technology can be used to improve assessment can be found in chapter 5.

### 4.3.1 E-Assessment Types

E-Assessment is often also called *Computer-Based Assessment (CBA)* or *Computer-Assisted Assessment (CAA)* and most people think that these terms mean the same but actually CBA and CAA are subgroups of E-Assessment. According to Higgins and Bligh (2006) these two E-Assessment forms can be distinguished by the number of processes that usually are automated, which is visualized in Figure 4.

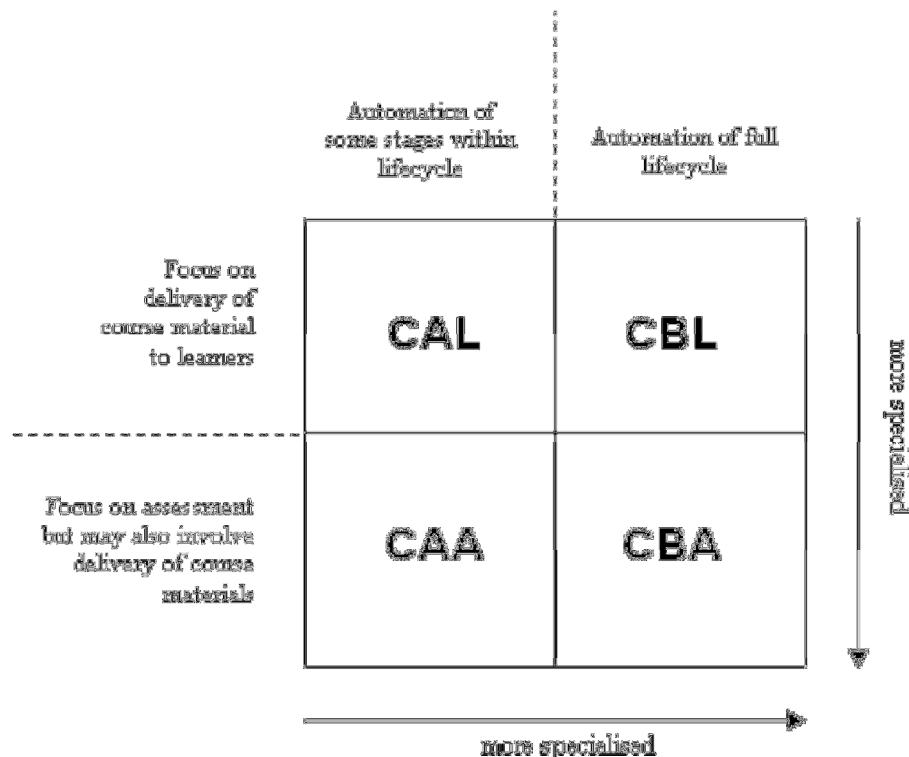


Figure 4: Relationships between CBA, CAA, CBL and CAL (Higgins & Bligh, 2006)

Computer-Based Assessment is the interaction between the computer and the user during an assessment process and as the name indicates, the whole process is based on computers and can only be realized with them. It is used for computer based tests where the assessment and feedback provision can be done by the computer. Computer-Assisted Assessment is more general. It includes any use of the computer during the assessment lifecycle and so it could cover the whole process. Sometimes in CAA the computer is never directly used for the assessment itself but plays a role in test analysis or reporting (AL-Smadi & Guetl, 2008).

Figure 4 also includes *Computer-Based Learning (CBL)* and *Computer-Assisted Learning (CAL)* which are subgroups of E-Learning and are focused on the delivery of course material to learners. As the figure shows the computer based types are more automated and specialised than the computer assisted ones, especially Computer-Based Assessment. (Higgins & Bligh, 2006)

### 4.3.2 Computer-Based versus Paper-Based Assessment

In the last years the use of computer based testing and assessment has increased a lot. Examples are fully automatic E-Learning courses, the state driver's license exam or job application exams. Before those years these exams were done in the classical paper based way with the same content and cognitive activities as today. Although experiment findings have shown that identical computer-based and paper-based tests do not produce the same results. This phenomenon is known as the "*test mode effect*". (Clariana & Wallace, 2002)

Based on these findings Higgins and Bligh (2006) tried to identify the advantages of CBA over traditional, paper based assessment. Generally its benefits can be classified into *practical* and *pedagogic* ones and the practical benefits are defined by the fact that CBA can be used to automate the assessment processes for a large amount of students without increasing the workload of teachers and tutors. For identifying the pedagogic benefits, 10 criteria were chosen for measuring the quality of assessment. These criteria are shown in Table 2 and furthermore, it is checked whether or not these criteria are met by CBA.

It can be seen that CBA has a pedagogic advantage in 7 of 10 criteria and in the remaining 3 criteria (fair, incremental and demanding) it is design-dependent and so has no disadvantage. Especially CBA is very powerful when a reliable and consistent assessment method is important because exactly the same assessment process will run for each submission and student. It also fits perfectly the needs of formative assessment because it offers the opportunity to efficiently run the assessment process multiple times to provide full continuous feedback. (Higgins & Bligh, 2006)

<b>Criterion</b>	<b>Application to CBA</b>
Valid	Will measure specified coursework aspects assuming good initial assessment design
Reliable	The same assessment process will run for each submission; consistency is absolute
Fair	Design-dependent: CBA has no inherent advantages
Equitable	The same assessment process will run for each submission; discrimination is non-existent
Formative	CBA provides a good opportunity to run assessment frequently throughout the learning process, and to provide multiple submissions with full feedback each time
Timely	CBA provides a good opportunity to run assessment frequently throughout the learning process
Incremental	Design-dependent: CBA has no inherent advantages
Redeemable	CBA is suited to allowing multiple submissions should the designer wish this
Demanding	Design-dependent: CBA has no inherent advantages
Efficient	Considerable time and other resource savings to be made; originally a motivator for CBA's development

Table 2: Application of Assessment Criteria to Computer-Based Assessment (Higgins &amp; Bligh, 2006)

The next section gives a short overview about assessment rubrics to understand the basic mechanisms of this flexible assessment tool. Rubrics are often used to assess written essays and scientific works and so they were also used in the developed Wiki of this thesis to group assess Wiki contents.

## 4.4 Assessment Rubrics

Rubrics are powerful tools to provide a quick, fair, efficient and consistent assessment for students. Moreover, they can also be used to increase the students' learning and to blur the distinction between instruction and assessment. Usually a rubric consists of two parts: a number of criteria that describe what counts in the assignment and a list of quality levels for each criterion to rate how good this criterion was achieved by the student. The list of

## Chapter 4: E-Assessment

criteria can also have multiple hierarchical levels if necessary. (Andrade, 2000) However, Figure 5 shows an example of a rubric with one criteria level that can be used to assess paper abstracts.

Criteria	Points Possible	Points Awarded	Comments
Listing (completeness, formatting)	5		
Abstract (depth/clarity)	15		
Analysis of Usefulness/Relevance to teaching and learning/instructional practice	15		
Clarity and Structure of paper/ Grammar and Expression in written review	5		
Sharing with/Presentation of Abstract to Class	10		
Total	50		

Figure 5: Example assessment rubric for paper abstracts (TLT, 2009)

Based on Andrade (2000), rubrics have multiple advantages. First, they are *easy to use and to explain* and *make the assignments' expectations very clear*. This is because usually rubrics are provided to the students at the beginning of an assignment and so the students exactly know what is expected by them. Furthermore rubrics *support learning processes* and the *development of skills*. This has been shown in one of her experiments where the students who have known the rubric have also received better grades.

As rubrics are only simple tables in the most cases they can also be mapped to electronic structures such as Excel sheets or relational databases. Based on this fact, several tools have been developed that help the teacher to create rubrics. Berger (2011) named some of these tools:

- Rubistar<sup>16</sup>
- Rubic Maker at Recipes4Success<sup>17</sup>
- Assessment Generators Tools at the Canadian Teacher<sup>18</sup>

<sup>16</sup> <http://rubistar.4teachers.org/>; visited on 21<sup>st</sup> of January 2012

<sup>17</sup> [http://myt4l.com/index.php?v=pl&page\\_ac=view&type=tools&tool=rubricmaker](http://myt4l.com/index.php?v=pl&page_ac=view&type=tools&tool=rubricmaker); visited on 21<sup>st</sup> of January 2012

- iRubric at RCampus.com<sup>19</sup>

The Wiki system that was enhanced as the practical part of this thesis also uses a rubric for group assessment. This is further explained in subsections 5.2.2 and 8.2.2.

## 4.5 Summary

Assessment is an integral part of all teaching, learning and feedback processes and is used to not only improve the students' learning activities but also to improve the effectiveness of teachers. To support these important processes modern information and communication systems are used to automate specific steps of it. This is also the main definition of E-Assessment and there can be identified two main motivations for the use of it, *practical* and *pedagogic* ones.

Assessment methods can be differentiated based on their strategy and their usage type. Crisp (2009) explained the strategies of assessment with his well-known assessment cycle. He identified *diagnostic assessment*, which is the assessment before learning and is used to define the prior knowledge of students, *formative assessment*, which is the assessment during learning and should give continuous feedback to students and *summative assessment*, which is the assessment after learning and is mainly used for student grading. Based on the usage type, *Peer-, Self-, Group-, Collaborative-, Teacher- and Automatic-Assessment* can be differed.

Due to the fact that the number of students increases each year, but the number of teachers and tutors nearly remains the same, new forms of technology enhanced assessment have evolved. These types of E-Assessment can be categorized into *Computer Based Assessment (CBA)* and *Computer Assisted Assessment (CAA)*. When using the first type the whole assessment process depends on the computers and when using the second type computers are just used to support the assessment process in any of its steps. Higgins and Bligh (2006) have tried to identify the advantages of CBA over traditional paper based assessment. For this they checked if CBA is able to meet 10 criteria that are used for measuring the quality of assessment. Findings suggest that CBA has a pedagogic advantage in 7 of 10 criteria and so can be seen as a more powerful assessment method than paper based assessment when designed properly.

---

<sup>18</sup> <http://www.thecanadianteacher.com/tools/assessment/>; visited on 21<sup>st</sup> of January 2012

<sup>19</sup> <http://www.rcampus.com/indexrubric.cfm>; visited on 21<sup>st</sup> of January 2012

## Chapter 4: E-Assessment

A quick, fair, efficient and consistent way to provide assessment and feedback in E-Learning environments is the use of rubrics. Usually such an assessment rubric consists of two parts: a number of criteria that describe what counts in the assignment and a list of quality levels for each criterion to rate how good this criterion was achieved by the student. There also exist online systems that can be used to create rubrics. Examples of such systems are Rubistar, iRubric, the rubric maker at Recipes4Success and the assessment generators tools at the Canadian Teacher (Berger, 2011). The next chapter focuses on ways how the principles from the last two chapters can be combined in order to combine CSCL with E-Assessment. This chapter also acts as a bridge between the theoretical the practical parts of this thesis.

## 5 Combining Computer-Supported, Collaborative Learning with E-Assessment

As stated in chapter 3, Computer-Supported, Collaborative Learning (CSCL) is a research area that describes approaches how collaborative learning can be supported with modern computer and information technologies. These computer-supported environments should help the learners in bringing together their individual points of view to create new collaborative content and knowledge (Ma, 2008). The most common approach of CSCL is Computer-Supported, Collaborative Writing, which is the process of supporting joint writing projects with the help of blogs, Podcasts, virtual online worlds and especially Wiki systems (Noel & Robert, 2004).

Another important part of learning is the assessment of it, especially its computer-supported form, E-Assessment. E-Assessment was defined in chapter 4 as all methods that are used to define, evaluate, rate, document and feedback learning processes before, during and after the teaching period with the help of modern technologies (Brahm & Seufert, 2007). A big part of today's research in these areas focuses on how E-Assessment can be used within CSCL environments, particularly Wiki systems.

The aim of this chapter is to reflect the current research work that is based on the combination of Wiki systems with E-Assessment methods, to identify its open issues and problems and to suggest solution approaches for them. This is done in the following subsections after stating the motivation and problem definition for this research area. Furthermore, this chapter is used as a theoretical input for the practical part of this thesis that starts with chapter 6.

### 5.1 Motivation and Problem Definition for Using E-Assessment Together with CSCL

According to AL-Smadi, Guetl and Chang (2011), assessment should be an integral part of every learning activity, which also includes collaborative learning processes. Thus, CSCL activities that get integrated in a course have to influence the assessment procedures and forms. Such integrated assessment activities have a high potential to increase the students' motivation and to attract their attention. One big problem of the assessment of collaborative work is the detection of individual contributions, which could lead to a lack of fairness as defined by the free-riding problem. However, Wiki systems are designed in a way that

they track all changes made to Wiki pages by users, which also makes it possible to visualize and analyse individual contributions. (AL-Smadi, Guetl, & Chang, 2011)

Generally the assessment of CSCL consists of two parts: the assessment of the *product*, which is the output of the collaborative activity, and the assessment of the *process*, which is the sum of individual contributions that lead to the product (Macdonald, 2003). Furthermore, she suggested the following guidelines for a successful integration of assessment into CSCL:

- In cases where CSCL related skills, such as IT or online communication skills are part of the course, they need to be assessed too.
- These additional skills have to be covered by the course material and the assessment of it, especially when the collaborative working processes cause additional work for the students.
- Assessment and collaborative online activities should always be linked together and the assessment aims and goals should also be shared with the students.

In fact, if a piece of work, like the collaborative creation of a Wiki page, gets assigned to a group of students without a planned assessment for it, the students will not work on it (AL-Smadi, Guetl, & Chang, 2011).

This indicates that the use of Wikis in education only makes sense when the Wiki activities get also assessed by teachers or by peers. Section 3.4.2, and especially its subsection 3.4.2.2, has shown that today's Wiki systems lack these assessment functionalities and so also lack motivational aspects for students. Moreover, these educational Wikis are mainly designed for teachers as knowledge repositories for their learning material and not for collaborative writing projects. This means that there are no built-in assessment and feedback mechanisms and also that some of them are very prone to the free-riding problem. The next section focuses on related work that is based on this problem definition.

## 5.2 Related Work

As stated in the previous section, the research in Computer-Supported Collaborative Learning focuses on understanding collaborative learning practices and on conditions for successful collaborative learning processes (Strijbos, 2011). As a result, current Wiki systems lack formative and summative assessment possibilities to support the students' learning processes. Furthermore, it is hard for the teacher to identify individual contributions and so the

free-riding problem is an actual issue in collaborative writing assignments. The remainder of this section presents actual research work for assessing individual contributions and for using rubrics and peer assessment methods in Wiki systems.

### 5.2.1 Assessing Individual Contributions in Collaborative Writing Projects

The work of Biuk-Aghai, Kelen and Venkatesan (2008) described a way for supporting the assessment process in Wiki systems with the help of visualizations. These visualizations should help to better understand the process and outcome of collaborative writing projects. In order to do and test this, they implemented the so called *TransWiki* system, which is an enhanced version of the MediaWiki software used for translations. During their experiments the research group was interested in three specific research questions: (1) How much has each student contributed to the final outcome? (2) What was the process of collaboration? (3) What was the depth of collaboration? To answer these questions, a visualization tool was build that used the data from the revision history. Such visualizations explain the process and outcome of the collaboration much better than the history that can be very unclear in case of long Wiki pages. The following listing shows some examples of these visualizations and explains how they can be used for answering the questions above:

- **Contribution summary graph**

This visualization (Figure 1) shows how much each student has contributed to a specific page or to the complete outcome and so can give an answer to research question (1).

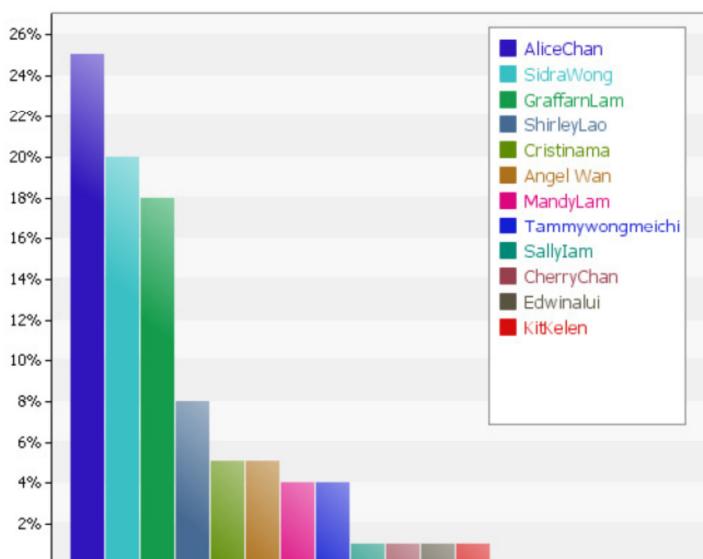


Figure 6: Example of a contribution summary graph (Biuk-Aghai, Kelen, & Venkatesan, 2008)

- **Analysis graph of all users**

Here it is shown how the process of creating a page was measured over time. This gives an answer to question (2) and is shown in Figure 7.

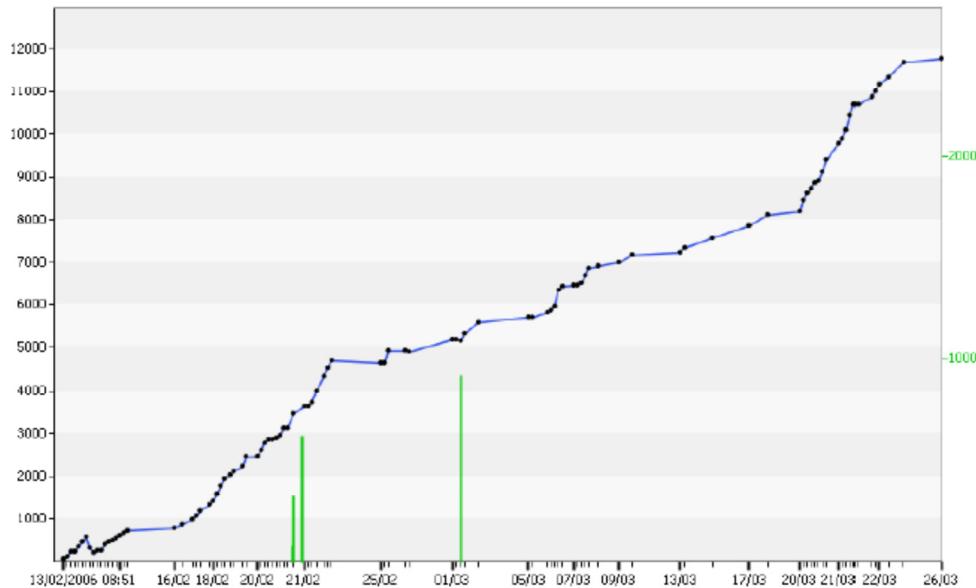


Figure 7: Example of an analysis graph (Biuk-Aghai, Kelen, & Venkatesan, 2008)

- **Text author graph**

This graph (Figure 8) displays the colour-coded content of a Wiki page. Each author is shown with a different colour, which gives an assumption of the collaboration depth and so answers research question (3).

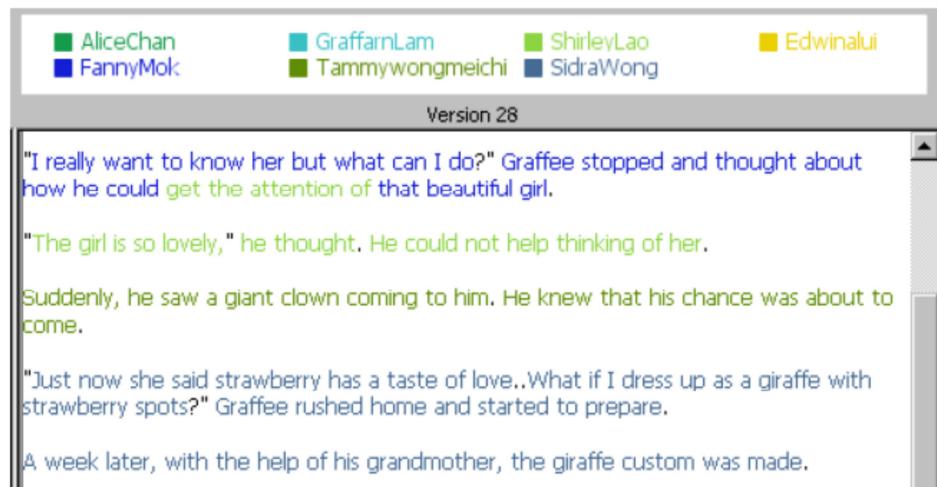


Figure 8: Example of a text author graph (Biuk-Aghai, Kelen, & Venkatesan, 2008)

Based on these visualizations Biuk-Aghai and his group were able to better assess the work of students where they observed that the quality of their translations has improved through the collaborative work within the Wiki (Biuk-Aghai, Kelen, & Venkatesan, 2008).

Another method of supporting the assessment of individual student contributions in Wiki systems has been explained by Khandaker and Soh (2010). They have designed and implemented a Wiki called *ClassroomWiki* that is based on the Java Spring Framework and a MySQL data backend. The innovative part of this system is that it builds a detailed student model based on all activities that happens in the Wiki during collaboration. These student models can be used by the teacher for assessing individual contributions or to intervene into the collaborative writing process in cases where the group work is not heterogeneous. The categories of such a model can be seen in Table 3.

Category	Tracked Information
Active use	All actions that change the content of Wiki pages such as adding, deleting or editing words.
Passive use	Actions where the Wiki is used but no content is changed. For instance, the number of logins or the number of times the revision history gets viewed.
Interaction	All interactions with other students to support the collaboration. For example, the total number and size of created discussion threads or replies.
Survey response	Responses to surveys or questionnaires such as the number of given peer reviews or reviews about the Wiki itself.
Evaluation	The teacher, tutor and peer evaluation scores received by the student of this model.

Table 3: Student model used by ClassroomWiki (Khandaker & Soh, 2010)

In addition, the authors of this work implemented a group formation mechanism in the Wiki, which is based on the student model above, to automatically build balanced student groups for collaborative writing assignments. Their experiment results have shown that these automatically built groups have better collaborative learning outcomes than randomly selected groups. (Khandaker & Soh, 2010)

### 5.2.2 Using Rubrics and Peer Assessment in Wiki Systems

Rubrics are powerful tools to provide a quick, fair, efficient and consistent assessment for students. Usually a rubric consists of two parts: a number of criteria that describe what counts in the assignment and a list of quality levels for each criterion to rate how good this criterion was achieved by the student. (Andrade, 2000)

## Chapter 5: Combining Computer-Supported, Collaborative Learning with E-Assessment

The work of Lai and Ng (2011) described a way of using such rubrics in a Wiki based course for student teachers to assess their collaboratively created artefacts. The aim of their study was to identify if rubrics can be used to integrate self assessment and peer assessment activities in Wiki based processes. Thus, they did an experiment with a class of student teachers where they asked them to create Wiki pages for providing learning material through collaboration within their group. Furthermore, they were asked to design a rubric and to self assess their Wiki pages with the help of it. In the last step of the experiment they were begged to use the rubrics of the other groups to peer assess their work. The concept of rubrics was new to most of the student teachers, so it was quite difficult for them to agree on criteria that can be applied to Wiki pages. However, the research findings suggested that they liked the use of rubrics for self assessment and peer assessment activities in Wikis because of the fair and consistent assessment guidelines that they provide. In addition the rubrics helped the students to better understand how well-designed Wiki pages should be structured. (Lai & Ng, 2011)

Another example of the use of Wiki rubrics for the assessment of collaborative writing has been stated by Jones, Kolloff and Kolloff (2010). In their paper they presented seven best practices for using Wikis for learning and assessment. One of these practices is the use of assessment rubrics. An example of such a rubric is shown in Figure 9.

CATEGORY	19-20	14-18	9-13	0-8
<b>1. Content</b>	Covers topic presented in-depth with details and examples. Subject knowledge is excellent. Good use of resources and references all information clearly.	Includes essential knowledge about the topic. Subject knowledge appears to be good. References information in most cases (two or fewer references missing).	Includes essential information about the topic but there are 1-2 factual errors. References for information are missing for three or more facts shared.	Content is minimal, OR there are several factual errors. No references for information provided.
<b>2. Organization</b>	Content is well organized, using headings (those given in the research guide) or bulleted lists to group related material.	Content uses headings or bulleted lists to organize, but the overall organization of topics appears flawed.	Content is logically organized for the most part.	There was no clear or logical organizational structure, just lots of facts.
<b>3. Attractiveness</b>	Makes excellent use of font, color, graphics, effects, etc. to enhance the presentation.	Makes good use of font, color, graphics, effects, etc. to enhance the presentation.	Makes use of font, color, graphics, effects, etc. but occasionally these detract from the presentation content.	Makes use of font, color, graphics, effects, etc. but these often distract from the presentation content.
<b>4. Contribution to the Group</b> (Evident in postings to the wiki and through the group discussion board.)	Contributes greatly to the development of the class wiki. Very active in the group discussions and in developing the wiki.	Contributes adequately to the development of the class wiki. Adequately active in the group discussions and in developing the wiki.	Contributes moderately to the development of the class wiki. Moderately active in the group discussions and in developing the wiki	Contributes minimally to the development of the class wiki. Very little or no activity in the group discussions and in developing the wiki.
<b>5. Accuracy</b>	No misspellings or grammatical errors. No HTML errors in wiki (e.g., broken links, missing images).	Two or fewer misspellings and/or grammatical errors. No more than two HTML errors in the student's contribution to the wiki.	Three misspellings and/or grammatical errors. No more than four HTML errors in the student's contribution to the wiki.	More than three errors in spelling or grammar. Five or more HTML errors in the student's contribution to the wiki.

Figure 9: Example rubric for assessment using Wikis (Jones, Kolloff, & Kolloff, 2010)

Such a rubric in combination with self assessment gives the students a better understanding of what is expected by their work, for example the required citation style (Jones, Kolloff, & Kolloff, 2010).

The work of Xiao and Lucking (2008) compared the impact of two types of peer assessment on the writing performance of students in a Wiki based learning scenario. Peer assessment was defined in subsection 4.2.2 as an assessment method where students provide their peers with grades and feedback (Bostock, 2000). In their experiment, Xiao and Lucking (2008) used a rating-only peer assessment method for one student group and a rating-plus-feedback method for the other group and compared the results. Their findings demonstrated that students who received ratings and feedback comments by their peers showed better improvements in their collaborative writing performance than the students that only received ratings. Moreover, their results showed that the students' peer-reviews are very similar to the ones that were given by the teachers and so can be very helpful for both sides, students and teachers. (Xiao & Lucking, 2008)

## 5.3 Identified Issues

The last section has shown that the combination of Computer-Supported Collaborative Learning and E-Assessment is a vital research area and that this topic is of high interest in the educational sector. However, this literature review has also illustrated that there are still open issues, especially in the educational use of Wiki systems. These identified issues are further explained in this section.

AL-Smadi, Hoefer and Guel (2011a) summarized the main problems that are faced when working with CSCL environments:

- **Lack of Awareness**

Group members should have task-related awareness and social awareness. Task awareness focuses on what the other group members have done, what they know about the current task and what they are planning to do next. On the other hand, social awareness deals with questions such as: Who is available for discussions or who needs help to do his or her contributions?

- **Coordination Problems**

In order to work collaboratively, the group members have to coordinate the assignment tasks. This includes the initial distribution of the tasks among the members, the sharing of knowledge and the creation of a common understanding of the

topics to maintain the group production function and well-being. (AL-Smadi, Hoefler, & Guetl, 2011a)

- **Communication Problems**

It is important that the CSCL environment offers a communication media for the users in order to avoid coordination problems.

- **Motivational Aspects**

As described in section 5.1, today's Wiki systems mainly lack of assessment possibilities, which is the most important motivational aspects for students. This problem is also mentioned by Strijbos (2011). He claims that the assessment activities in CSCL environments are mostly only summative, are only designed and conducted by the teacher and are only focused on the outcome and not the process of the work.

Above these points, subsection 3.4.2.2 has shown that some of the Wikis lack of suitable permission settings for collaborative writing assignments. This means that it is often only possible to set global permissions for users on all the Wiki pages, so each student group can see the work of the other groups. Here it would be necessary to assign specific ranges to the particular groups where they can view, create and edit only their pages. Another problem that was also described in the previous section is the assessment of the individual contributions in collaborative writing projects. If this is not mentioned properly, the risk of free-riding occurs where lazy students benefit from the group mark. The next section gives possible solutions approaches for these identified issues.

## 5.4 Solution Approaches

This section is based on the problems that were identified in the previous sections and gives possible solution approaches for them. It can be seen as a summary of the theoretical ideas that were used to implement the Wiki features which are described in the practical part of this thesis (see chapter 6). The solution approaches are presented in the following listing in relation to the problems from section 5.3.

**Lack of Awareness:** As a Wiki tracks all the page revisions and saves them in a log file or in a database, these revisions can be visualized in a history table for all the Wiki pages that belongs to a specific student group. This table could show all the revisions together with the intention and the importance of each revision to maintain the students' task awareness. To support social awareness the Wiki could provide an online user list to let the students

know who would be available for discussions and a contribution summary graph as shown in subsection 5.2.1. (AL-Smadi, Hoefler, & Guetl, 2011a)

**Coordination and Communication Problems:** In order to tackle coordination and communication problems, the Wiki has to provide a reliable communication medium. This could be a live chat or a discussion forum on the level of pages. The second solution has the big advantage that the students can also discuss when they are not online at the same time.

**Motivational Aspects:** To highly increase the motivation of students in collaborative writing assignments, the Wiki needs integrated assessment possibilities. Here it is important that also formative assessment and feedback is provided, which could be implemented with short self- and peer assessment activities after each revision. Nevertheless, also summative assessment is important and can be easily provided via assessment rubrics. (Jones, Kolloff, & Kolloff, 2010) Furthermore, the teachers should have access to the page revision histories to assess the writing process and not only the writing outcome.

**Group Work Support:** Here it is fundamental that the Wiki provides areas for student groups where only the group members can see and edit the pages (like the principle of namespaces in MediaWiki). These areas also need to be connected to an assignment to make group wide peer assessments possible. As an addition, the teacher should also be supported by helpful visualizations as demonstrated by Biuk-Aghai, Kelen and Venkatesan (2008) to fairly assess individual contributions and to tackle the free-riding problem.

## 5.5 Summary

According to AL-Smadi, Guetl and Chang (2011), assessment should be an integral part of every learning activity, which also includes collaborative learning processes. Thus, CSCL activities that get integrated in a course have to influence the assessment procedures and forms. Such integrated assessment activities have a high potential to increase the students' motivation and to attract their attention. One big problem of the assessment of collaborative work is the detection of individual contributions, which could lead to a lack of fairness like the free-riding problem.

The work of Biuk-Aghai, Kelen and Venkatesan (2008) described a way for supporting the assessment process in Wiki systems with the help of visualizations. These visualizations should help to better understand the process and outcome of collaborative writing projects. In order to do and test this, they implemented visualizations in a Wiki system to answer specific research questions: (1) How much has each student contributed to the final outcome? (2) What was the process of collaboration? (3) What was the depth of collabora-

tion? Their findings have shown that such visualizations can explicitly increase the assessment quality and can help to tackle the free-riding problem. Another example of the use of Wiki rubrics for the assessment of collaborative writing has been stated by Jones, Kolloff and Kolloff (2010).

Another research field in this area focuses on the use of rubrics to assess the process and outcome of CSCL assignments. The work of Lai and Ng (2011) described a way of using such rubrics in a Wiki based course for student teachers to assess their collaboratively created artefacts. Their research findings suggested that they liked the use of rubrics for self assessment and peer assessment activities in Wikis because of the fair and consistent assessment guidelines that they provide. In addition the rubrics helped the students to better understand how well-designed Wiki pages should be structured.

There are four main issues that today's CSCL environment lack when they are used in educational settings.

**Lack of Awareness:** As a Wiki tracks all the page revisions and saves them in a log file or in a database, these revisions can be visualized in a history table for all the Wiki pages that belongs to a specific student group. This table could show all the revisions together with the intention and the importance of each revision to maintain the students' task and social awareness.

**Coordination and Communication Problems:** In order to tackle coordination and communication problems, the Wiki has to provide a reliable communication medium. This could be a live chat or a discussion forum on the level of pages.

**Motivational Aspects:** To highly increase the motivation of students in collaborative writing assignments, the Wiki needs integrated assessment possibilities. Here it is important that also formative assessment and feedback is provided, which could be implemented with short self- and peer assessment activities after each revision.

**Group Work Support:** Here it is fundamental that the Wiki provides areas for student groups where only the group members can see and edit the pages (like the principle of namespaces in MediaWiki).

The next chapter acts as the first chapter of the practical part of this thesis and describes the requirements and the design of the developed Wiki prototype.

## 6 Requirements and Design of the Prototype

The purpose of the previous chapters of this thesis was to give a theoretical basis and background for the practical part that starts with this chapter. The most important aspects of Technology Enhanced Learning (TEL), E-Education, Computer-supported, Collaborative Learning (CSCL), Wiki systems and E-Assessment have been described. Moreover, actual research questions and problems in the field of combining CSCL with E-Assessment, which is the main part of this thesis, have been analysed in chapter 5. Here especially four issues were identified: the lack of awareness, coordination problems, communication problems and motivational aspects, which includes also the lack of assessment activities.

The aim of the practical part of this thesis is to present and implement a solution approach for these identified issues of combining CSCL with E-Assessment. Thus, a Wiki system has been enhanced with assessment methods and other tools to support teachers and students in collaborative writing activities. This chapter gives an overview of the project idea, the requirements, the conceptual architecture, the chosen Wiki framework and the design of the developed Wiki prototype. Some parts of the practical part of this thesis, especially screenshots of the prototype, have already been published as official ALICE deliverables, but it is especially mentioned if a paragraph or a figure has not been created by the author alone.

### 6.1 Project Idea and Description

The aim of this project is to enhance an existing Wiki system with flexible assessment methods to increase the engagements of students and learners. This means that the system should not only be usable like a typical Wiki system with all its advantages, but also be enhanced with integrated assessment and feedback features and other motivational aspects. These enhancements should support students and teachers in collaborative writing assignments and should tackle the open issues that were identified in section 5.3. Furthermore it is a main goal of this project that the Wiki should be usable in two ways, as a standalone Web application and as an integrated Web tool in existing LMSs. The project has been processed in three phases: first the requirements and some mock ups for the user interface have been defined, then the stand alone application has been implemented and in the last phase the integration aspects have been developed. During the phases 2 and 3, internal and external experimentations have been conducted to ensure that all implemented features are tested properly and to get direct input from the users in order to improve the Wiki.

The practical part of this thesis has been developed as part of the ALICE project, which stands for *Adaptive Learning via Intuitive/Interactive, Collaborative and Emotional Systems*. ALICE is a project in the ICT (Internet and Communication Technologies) area and is co-funded by the *European Commission under the Seventh Framework Programme for Research and Technological Development (FP7)*<sup>20</sup>. The aim of ALICE is to increase the engagement of students and learners in E-Education, which is defined by the following aspects (ALICE, 2009):

- Challenge: the learning material needs to be challenging to be attractive
- Empowerment: the learner has to control the learning experience and not vice versa
- Social identity: the E-Learning system should not give the students the feeling that they are learning alone
- Interaction: there should be a dynamic interaction between the student and the learning material

The ALICE project consists of six work packages that are assigned to the five participants: University of Technology Graz (TUG), Coventry University<sup>21</sup>, University of Catalonia<sup>22</sup>, MOMA<sup>23</sup> and CRMPA<sup>24</sup>. The Wiki system that is described in the remainder of this thesis is part of work package 5 (WP5) that deals with new forms of assessment and collaborative learning. The tools implemented as part of these work packages will all be integrated in the IWT (Intelligent Web Teacher) portal<sup>25</sup>. The AEMT<sup>26</sup> group of IICM, TUG has been responsible for WP5 and the author of this thesis has been in charge of the design and the implementation of the prototype. The next section will further explain the requirements that have to be fulfilled to achieve these project goals. (ALICE, 2009)

## 6.2 Requirements

As described in the last section, the aim of this project is to enhance an existing Wiki system with flexible assessment methods to increase the learning experience in collaborative writing assignments. Based on these project goals, the requirements have been determined in order to choose an appropriate Wiki system and to enhance it according to the project

---

<sup>20</sup> <http://cordis.europa.eu/fp7/>; visited on 2<sup>nd</sup> of February 2012

<sup>21</sup> <http://wwwm.coventry.ac.uk/>; visited on 3<sup>rd</sup> of February 2012

<sup>22</sup> <http://www.uoc.edu/>; visited on 3<sup>rd</sup> of February 2012

<sup>23</sup> <http://www.momanet.it/>; visited on 3<sup>rd</sup> of February 2012

<sup>24</sup> <http://www.crmpa.it/>; visited on 3<sup>rd</sup> of February 2012

<sup>25</sup> <http://elearning.diiima.unisa.it/>; visited on 3<sup>rd</sup> of February 2012

<sup>26</sup> <http://www.iicm.tugraz.at/about/Homepages/cguetl/team>; visited on 9<sup>th</sup> of April 2012

## Chapter 6: Requirements and Design of the Prototype

needs. The remainder of this section lists these requirements and is divided into two subsections, one describing the functional requirements and the other describing the non-functional requirements.

### 6.2.1 Functional Requirements

The requirements listed in this subsection are based on the project idea presented in the last section and on the theoretical analysis that has been stated in chapter 5. However, also some of the requirements were added during the implementation process based on experiment findings and the ALICE use case and scenario descriptions. The experiment findings are further explained in section 8.3. The remainder of this subsection is divided into two parts, the tool requirements and the integration requirements. The tool requirements describe the specific features of the Wiki and the integration requirements are based on the flexible usage scenarios of the Wiki, either as a standalone Web application or integrated in a LMS such as IWT or SOFIA.

#### 6.2.1.1 Tool Requirements

The following listings describe the different components of the functional tool requirements: assignment authoring, assessment activities, teacher views and student views. These requirements indicate the set of functions the Wiki should render.

##### Assignment Authoring

- User accounts
  - Users are defined by a unique username used for the login, a display name and an e-mail address
  - Create, read, update, delete (CRUD operations) and activate user accounts
- User groups
  - Users can be part of one or more user groups
  - One special user group for teachers and tutors
  - CRUD operations for user groups and their assigned users
- Namespaces
  - User groups and teachers can be assigned to one or more working areas within the Wiki (= namespaces)
  - Only assigned user groups can view, create and edit pages in their namespaces
  - Teachers have full access to their namespaces (includes delete-operations)
  - CRUD operations for namespaces and their contained pages

## Chapter 6: Requirements and Design of the Prototype

- Assignments
  - Namespaces can be assigned to one or more assignments to group them
  - Assignments are defined by a unique name, the scheduling dates for the group assessment phase and the mode of the internal peer assessment (see assessment requirements)
  - CRUD operations for assignments and their contained namespaces

### Assessment Activities

- Internal peer assessment
  - Used to assess changes to Wiki pages made by other group members
  - Can be done to the latest change before editing a page and can be configured by the teacher to be optional, mandatory, based on an intention (e.g. page creations) or based on a revision number (e.g. every 5th revision)
  - Is defined by an obligatory rating (0 – 5) to determine the importance of the change and an optional comment to give suggestions to the peers
  - Is shown in the Wiki page history to provide continuous feedback
- Self assessment
  - Used to assess own changes when editing a page (is mandatory)
  - Is defined by an obligatory rating (0 – 5) to determine the importance of the own change regarding the whole group work, an obligatory selection of pre-defined intentions for this change (e.g. edited paragraph) and an optional comment to give further notes
  - Is shown in the Wiki page history inform peers about the intention and the importance of the own changes
- Group assessment
  - Used to assess the output of groups on the level of the whole namespace or on the level of single pages
  - Can be done by students for self and peer assessment and teachers/tutors
  - Is done using an assessment rubric that provides a mandatory rating (0 – 5) and an optional comment for each criterion (see subsection 5.2.2)
  - Each rubric criterion has its own description and weighting that is shown to the user
  - Group assessment can be configured to assign specific groups to students (also the own group is possible) or to let each student peer assess all other groups of the assignment

## Chapter 6: Requirements and Design of the Prototype

- Teacher assessment and the averaged assessment from the student peers is shown to the user with all given comments in order to receive feedback for improvements
- Feedback of the group assessment phase is anonymous, only teacher see which student provided which assessment
- Inline Teacher Feedback
  - Used by the teacher to tag parts of Wiki pages with different colours in order to give specific feedback (e.g. check spelling)
  - Tagged feedbacks can be done for the actual page revision or based on feedbacks of previous revisions
  - Students can view the feedback when editing a page together with the colour key (e.g. blue means check spelling)

### Teacher Views

- Assignment overview
  - Contains a contribution visualization that shows each author of a Wiki page with a different colour to know who did what (see subsection 5.2.1)
  - Within this visualization it is also possible to navigate through all the revisions of a page (manually or automatically like a slideshow). This also shows the given internal peer assessments
  - Provides analysis graphs that show the collaborative creation process of a page (see subsection 5.2.1)
  - Contains a table with quantitative information (e.g. the number of letters)
  - All these information can be shown based on the whole group or a single student and on the level of all pages or a single page
- Grading schema
  - Calculates an automatic grade based on the averaged student group assessments and the teacher assessment
  - Student and teacher assessments are both weighted with 50% for the automatic grade, but this can be changed
  - Automatic grades can be saved for the whole group or for single users or can be manually changed if necessary

### Student Views

- Wiki functionalities
  - Login with username and password (is saved as a hashed value in the database)

## Chapter 6: Requirements and Design of the Prototype

- Self-registration of user accounts with the use of captchas to avoid automatically created spam accounts
- Possibility to change the password
- WSIWYG editor for Wiki pages that support the attachment of files, the integration of mathematical LaTeX formulas and syntax highlighting for code blocks
- Discussion forum on the level of pages to support group intern communication
- Assignment home page
  - Action feeds that show the Wiki page histories (can be filtered to show only the actions that occurred since the last own edit or login action)
  - Online user list to know who would be available for discussions
  - Contribution summary graphs on the level of the own group or all groups (see subsection 5.2.1)
- Assessment features
  - Visualization of the differences between two revisions on the level of words (shows added, removed and edited text and styles)
  - Visualization of the three phases of an assignment (contribution, assessment and enhancement) to show if group assessments have led to enhancements
  - Interactive view of the Wiki page structures based on the linking
  - Group assessment environment to navigate to the work of the groups that have to be assessed (see requirements for assessment activities)

### 6.2.1.2 Integration Requirements

The last subsection described the general tool requirements that have to be fulfilled for the standalone and the integrated version of the Wiki. The following listing shows the requirements that are needed to fully integrate the Wiki within an LMS:

- General aspects
  - The Wiki has to be usable with any number of LMSs
  - As part of this project the Wiki has to be fully integrated with IWT and the SOFIA system (*Service-Oriented Flexible and Interoperable assessment*) where parts of it has been implemented and designed as part of the author's master project (AL-Smadi & Guel, 2011)
- Login
  - The LMS users are directly logged in to the Wiki via an SSO (Single Sign-on) mechanism

## Chapter 6: Requirements and Design of the Prototype

- The Wiki receives the user data via a data provider plugin that is working with Web services of the LMS instead of the database (see section 6.5)
- Assignment authoring
  - Wiki provides an Web service interface to author assignments within the LMS
  - This interface offers CRUD operations for assignments and assigned user groups
- Grading
  - The teacher can use a special page in the Wiki to assign summative grades to students for different concepts that are defined by the LMS
  - These grades are saved in the Wiki database and are passed back to the LMS via provided Web services

### 6.2.2 Non-Functional Requirements

As opposed to the last subsection that has given an overview of the functional requirements, the aim of this subsection is to further explain the non-functional requirements. These requirements are not based on specific software use cases, but rather covering more general aspects and constraints. The following listing describes three parts of these requirements, the general ones, the technical ones and psychological ones, that also include the experimentations:

- General aspects
  - Usage flexibility by ensuring that the Wiki can be used as a standalone Web application or integrated within a LMS such as IWT or SOFIA
  - Increased performance and interactivity by the use of AJAX (Asynchronous JavaScript and XML)
  - Self-explaining usability via the use of simple controls
  - Interoperability and low costs by the use of open-source and cross-platform frameworks, database managements systems and control toolkits
  - Security via obligatory login mechanism and anonymous assessment activities
- Technical aspects
  - The Wiki has to be implemented based on the Microsoft .NET framework because also the LMSs where the Wiki will be integrated are based on it
  - Only free and open-source software with a good documentation can be used

## Chapter 6: Requirements and Design of the Prototype

- Important user actions get saved to the database to have the possibility to extract usage patterns from this data (e.g. save the login and logout timestamps to know how long the users stay online on average)
- Implementation of controls for a reusable design (e.g. all assessment and feedback activities use the same flexibly designed control)
- Complete XML style documentation of the code
- Use of different namespaces to divide the system into several packages (especially new Wiki extensions)
- New classes, Web pages and database tables have the prefix “ALICE\_” to be easily recognized
- Suitable database session and connection management to make the system be useable by a lot of clients at the same time
- Exception and error management at all code blocks
- GUI (graphical user interface) design is uniform (master pages), role-based (admin, teachers and students) and with English texts
- Psychological aspects
  - Experimentations to test all functionality and usability aspects in order to increase the quality of the Wiki (see section 8.3)
  - Creation of a promotional marketing video to describe the main functionalities and benefits of the Wiki (see Appendix: CD-ROM)

Based on these requirements, a conceptual architecture has been designed that shows the different components of the system and how they work together. This architecture is described in the next section.

### 6.3 Conceptual Architecture

As stated in subsection 6.2.2, the Wiki should not only be usable as a standalone Web application, but also be applicable as a subsystem of an LMS such as IWT or SOFIA. This separates the system architecture into two parts. The first part is the enhanced Wiki system itself to achieve the functional requirements from subsection 6.2.1 and the second part is the LMS where the Wiki is integrated into. This can be seen in Figure 10 which is an adapted version of the conceptual architecture that can be seen in AL-Smadi, Guetl, Caballe and Dunwell (2012).

For the sake of the main part of this project, the standalone Web application, the Wiki gets separated into two components. The first component is the already provided

## Chapter 6: Requirements and Design of the Prototype

Wiki engine that offers all typical Wiki functionalities such as user and group management, file and document management and revision and action logging. The second component is designed as all the enhancements and extensions that are implemented to fulfil the requirements such as the assignment authoring and management, the assessment and feedback features and the visualization and graph tools.

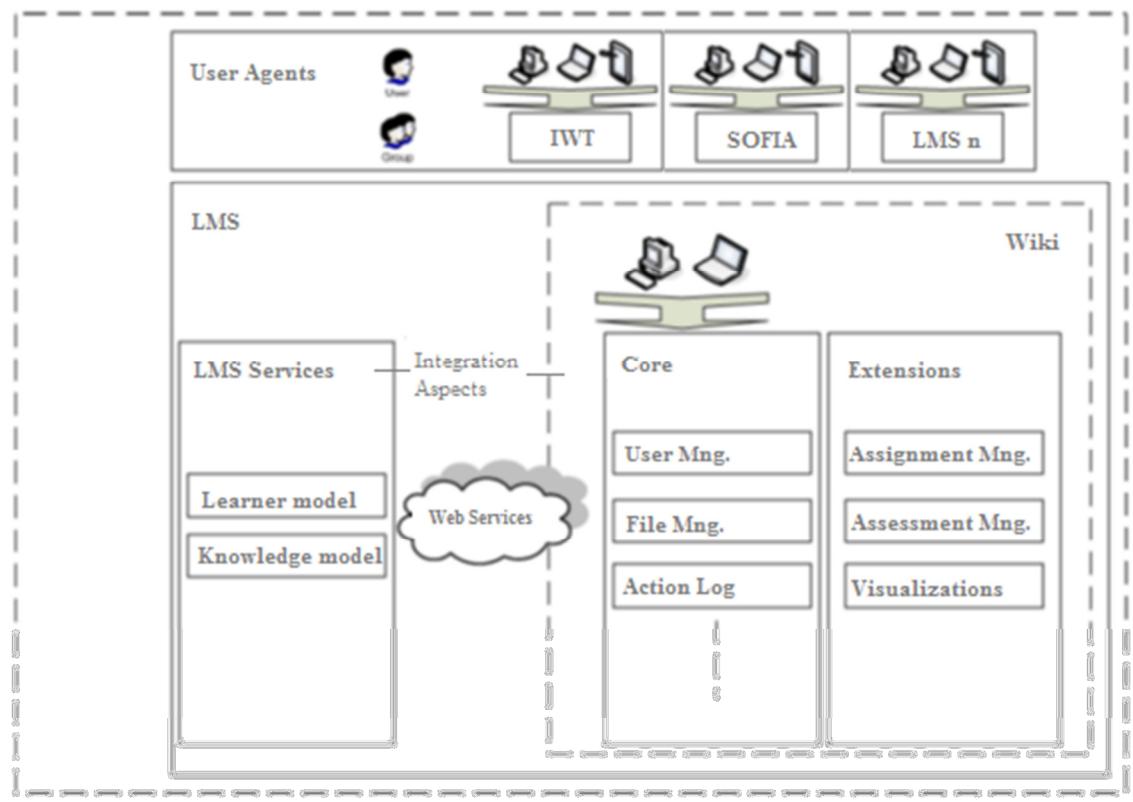


Figure 10: Conceptual architecture (AL-Smadi, Guetl, Caballe, & Dunwell, 2012)

The integration aspects of this project cover the following development tasks:

- Automatic login to the Wiki via a Single Sign-on (SSO) mechanism
- Assignment authoring via a provided Web service interface
- Update of the learner and knowledge model after the grading process via the use of Web services

As depicted in Figure 10, the user agents can access the Wiki either directly through its Web user interface or through a LMS such as IWT or SOFIA. Based on the role (student, teacher, tutor and admin) and the given permissions, the user will automatically be redirected to the functions that are available to him or her. On the basis of this architecture and the requirements, some potential Wiki frameworks have been tested and compared in order

to find the best one for the necessary enhancements. The considerations regarding these different frameworks are described in the next section.

## 6.4 Considerations Regarding the Wiki Software

One of the most important decisions in the design phase was to choose an appropriate Wiki system that can be enhanced in order to fulfil the requirements mentioned in the last section. An extensive Web research has shown that there is a lot of Wiki software available to do this. This number of possible Wiki frameworks gets constricted by the functional requirements (see subsection 6.2.2) and further limitations that ensure a successful development process:

- The Wiki architecture has to be easily extendible to achieve the necessary enhancements (e.g. with a plugin system)
- Only an often used Wiki with an active user community can be used to get support when needed
- The Wiki software should be similar to MediaWiki because most users know the look-and-feel of Wikipedia

On the basis of these requirements and listings the range of choices has been decreased a lot. Furthermore CSharp-Source (2010), which is an overview of free and open-source Microsoft .NET software, suggests the following Wiki systems that would fit the requirement profile:

- **FlexWiki<sup>27</sup>**

FlexWiki is open-source Wiki software developed by Microsoft. It offers all standard Wiki functionalities and offers a very good software design. It is also one of the few open-source Wikis that provides a build-in scripting language that can be used to add dynamic functionality to Wiki pages which makes it very powerful. However, the official development and support stopped in 2009 which does not make it an option for this project.

- **MindTouch DekiWiki<sup>28</sup>**

MindTouch DekiWiki is an open-source Wiki and Web mashup platform. It completely separates its business logic (based on the .NET framework) from its Wiki frontend (written in PHP), where all the functions of the business logic are provid-

---

<sup>27</sup> <http://sourceforge.net/projects/flexwiki/>; visited on 18<sup>th</sup> of February 2012

<sup>28</sup> <http://developer.mindtouch.com/>; visited on 18<sup>th</sup> of February 2012

ed by a REST (Representational State Transfer) Web service API (Application Programming Interface). This design makes it very flexible, but also very hard to enhance because Asp.NET controls and pages can only be integrated with a high effort.

- **ScrewTurn Wiki<sup>29</sup>**

ScrewTurn Wiki is a widely open-source Wiki system. It has a large user community that also develops a lot of plugins that can be integrated in the Wiki at runtime via its flexible plugin interface. With this plugin mechanism it is easy to extend the system with functionalities like a Single Sign-on integration. ScrewTurn is very similar to MediaWiki and also supports namespaces for pages and a WYSIWYG editor for the creation of Wiki pages. (CSharp-Source, 2010)

Based on these facts, ScrewTurn has been chosen as the Wiki framework that forms the basis of this project. Especially its flexible plugin system, the big user community and its similarity to MediaWiki have supported this decision. A detailed explanation of the architecture and features of ScrewTurn Wiki can be found in the next section.

## 6.5 Technical Aspects of ScrewTurn Wiki

ScrewTurn Wiki is based on Asp.NET 3.5 and usually runs with a simple file-based data backend, but also can be used with Microsoft SQL Server (Express) or MySQL via the use of plugins. It is even possible to mix different data storages in one ScrewTurn instance with the help of data providers which will be further explained in this section. (ScrewTurn, 2012)

The aim of this section is to give an overview of the technical aspects of the ScrewTurn Wiki software. Thus, first the system design is explained and then the ideas behind the ScrewTurn data providers, which are a central part of its architecture, are illustrated.

### 6.5.1 System Design

The ScrewTurn Wiki system is separated into several components that can be seen in Figure 11. The figure shows that the Wiki Web application uses the business logic components to present the Wiki pages to a browser. These functionalities can be extended by a flexible plugin system where the plugin framework acts as a middle ware between the

---

<sup>29</sup> <http://www.screwtturn.eu/>; visited on 18<sup>th</sup> of February 2012

plugins and the Wiki itself. The remainder of this subsection describes the Wiki components in detail.

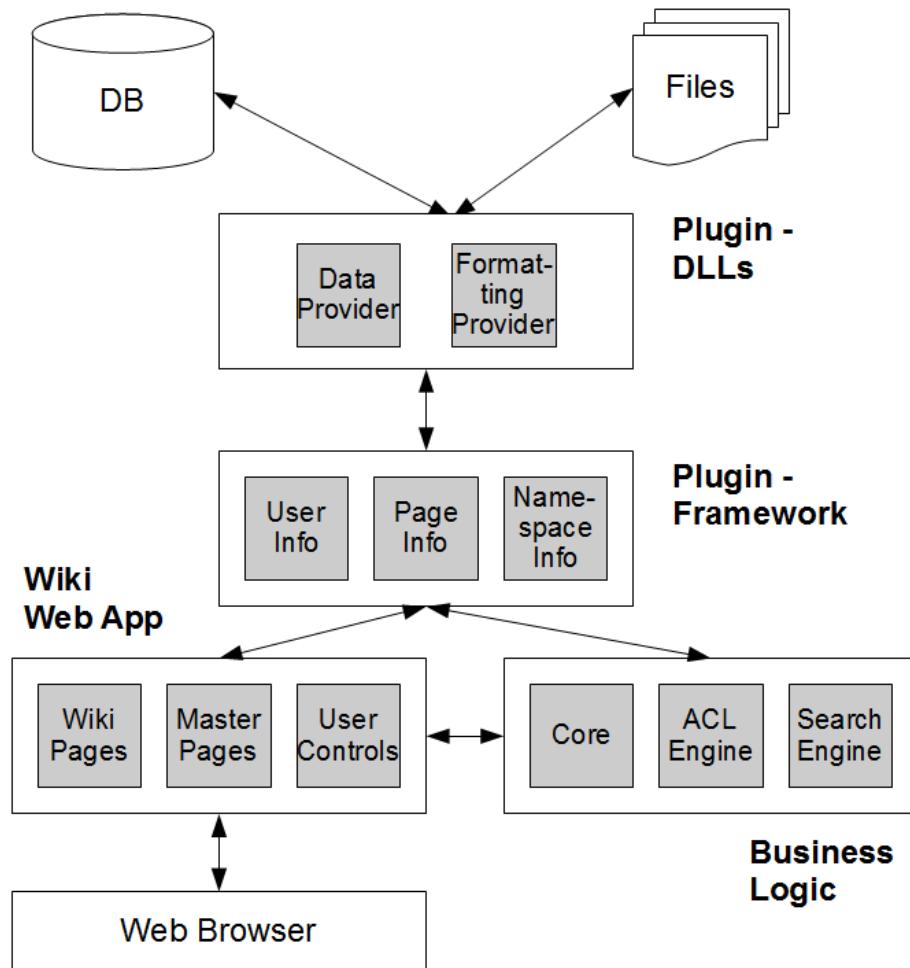


Figure 11: ScrewTurn Wiki system design

### Wiki Web Application

The Web application component is the Wiki itself. It contains all Asp.NET Web pages, user controls, JavaScript code files, CSS (Cascading Style Sheet) files and visual resources. The Wiki uses two different master pages, one for normal pages and one for admin pages, to ensure a consistent design of the Web pages. The functionalities of these Web pages are further explained in section 8.2.

### Core

The core assembly of ScrewTurn Wiki implements the business logic that is needed by the Web application. This includes main business objects and static classes that can be used like

## Chapter 6: Requirements and Design of the Prototype

interfaces to prevent direct interactions between the Web application and the data providers. The following listing shows the most important classes of this type:

- *AuthChecker*: used to check ACL permissions
- *Cache*: used to access the cache to minimize database interactions and to speed up the application
- *Content*: allows to retrieve specific revisions of Wiki pages
- *DiffTools*: used to calculate the difference between two revisions of a Wiki page
- *FilesAndAttachments*: contains methods to manage files and attachments
- *FormattingPipeline*: used to transform Wiki mark-up into HTML
- *Log*: used to log actions to the system log in the database
- *Pages*: handles all operations to retrieve and to create pages and namespaces
- *SearchTools*: used to access the build-in search engine
- *SessionFacade*: allows access to session based information like the username of the currently logged-in user
- *Settings*: contains methods to retrieve general settings like the database connection string
- *Tools*: offers utility functions that are needed in several parts of the applications
- *Users*: handles all methods to create and to retrieve users and user groups

### SearchEngine

The SearchEngine assembly contains all classes that are used to search through the Wiki. Generally, this search engine can be used to find Wiki pages based on phrases (also the page content gets searched) and is accessed via the *SearchTools* class.

### AclEngine

The ACL (Access Control List) engine handles the whole permission system of ScrewTurn. It is used to save an ACL entry to the database and to check if a *subject* has the necessary permissions to perform an *operation* on a *resource*. In ScrewTurn terms, a subject can be a user group or a single user, an operation can be an activity such as read or write and a resource can be a page, a namespace, an upload directory or a global resource. Table 4 shows the possible operations that can be granted or denied for a namespace. This is similar to the ones of pages and global resources. (ScrewTurn, 2012)

The permissions of a namespace are inherited by its pages and the permissions of a user group are inherited by its users, but of course can also be overridden if necessary. To

## Chapter 6: Requirements and Design of the Prototype

manually check permissions in code, the *AuthChecker* class of the Core assembly can be used.

Action	Description
<code>Full Control</code>	Full control on the namespace
<code>Read Pages</code>	Read pages
<code>Modify Pages</code>	Edit pages
<code>Create Pages</code>	Create new pages
<code>Delete Pages</code>	Delete, Rename pages
<code>Manage Pages</code>	Create, Edit, Delete, Rename pages
<code>Read Page Discussions</code>	Read page discussions
<code>Post Messages in Page Discussions</code>	Post messages in page discussions
<code>Manage Page Discussions</code>	Edit, Delete other users' messages in page discussions
<code>Manage Categories</code>	Modify category bindings of pages, create and delete categories
<code>Download Attachments</code>	Download page attachments
<code>Upload Attachments</code>	Upload page attachments
<code>Delete Attachments</code>	Delete, Rename page attachments

Table 4: ACL operations of ScrewTurn Wiki for namespaces (ScrewTurn, 2012)

## Plugins

The Plugins component contains all provider plugins that extend the functionality of the Wiki (e.g. for using external data storages such as SQL Server or MySQL). The principle of Wiki providers will be further explained in the next subsection. Also the communication with other systems via Web services is implemented as plugins.

### PluginFramework

The PluginFramework contains the classes of the transfer objects that are used to exchange data between the core assemblies and the different plugins. This has the big advantage that the core is independent from the concrete plugin implementations because plugins of the same type have to use the same transfer objects. The most important classes of this type are:

- *FileDetails*: represents an attachment
- *Message*: represents a thread or post in the discussion forum
- *NamespaceInfo*: holds the properties of a namespace
- *PageContent*: represents a specific page revision
- *PageInfo*: contains the metadata of a page
- *UserGroup*: represents an user group
- *UserInfo*: represents a single user

## Chapter 6: Requirements and Design of the Prototype

These classes are directly mapped to the corresponding database table which is further described in subsection 7.2.2. The next subsection explains the flexible provider plugin system of ScrewTurn.

### 6.5.2 Wiki Providers

One powerful feature of ScrewTurn Wiki is the provider concept. A provider is a DLL assembly that can be installed in the Wiki at runtime to do some work. In most cases these providers are used to fetch data, but they can also be used for other tasks, e.g. formatting the content of Wiki pages based on specific rules. The big advantage here is that there can be installed as many providers as necessary. This means that data can be fetched from different sources (e.g. databases, files, Web services) at the same time. Figure 12 shows the different types of providers and how they can be managed in the system by the administrators. (ScrewTurn, 2012)

The screenshot shows a web-based administration interface for managing providers. At the top, there is a navigation bar with radio buttons for selecting provider types: Pages Providers (selected), Users Providers, Files Providers, Cache Providers, and Formatter Providers. Below the navigation is a table with columns: Name, Ver., Author, and Update Status. One row is visible, showing "Local Pages Provider" with version "3.0.0.220", author "ScrewTurn Software", and update status "n/a". There is a "Select" button next to the status column. Below the table is a button labeled "Auto-update Providers". Under the heading "Default Providers", there are four dropdown menus: "Pages Provider" set to "Local Pages Provider", "Users Provider" set to "Local Users Provider", "Files Provider" set to "Local Files Provider", and "Cache Provider" set to "Local Cache Provider". Below these dropdowns is a "Save" button. The next section is titled "Providers DLLs Management" with a sub-section "Upload new DLL". It features a file input field, a "Browse..." button, and an "Upload" button. Below this is a dropdown menu labeled "- Select and delete -" and a "Delete" button. A note at the bottom states: "Note: removing a DLL won't disable the Providers it contains until the next wiki restart, but uploading a new DLL will automatically load the Providers it contains."

Figure 12: Wiki providers management screen (ScrewTurn, 2012)

For developing a new provider, a specific interface needs to be implemented. In case of a new pages provider this would be the *IPagesStorageProviderV30* interface which contains methods for managing namespaces, pages, page discussions, categories and search engine based tasks. In the course of this project, providers for fetching data from MySQL databases, for obtaining Single Sign-on (SSO) authentications and for formatting programming code blocks and mathematical LaTeX formulas were used. (ScrewTurn, 2012)

The next section gives an overview of the most important design considerations based on the results of the previous sections.

## 6.6 Design Considerations

The previous parts of this chapter have already explained the idea, requirements and considerations of this project. Based on these thoughts a decision for ScrewTurn Wiki has been achieved and the technical aspects of ScrewTurn Wiki have been explained. The aim of this section is to present a design on the basis of these results. Thus, the remainder of this section specifies the most important design considerations such as the data representation of the assignment authoring and management process and the assessment and feedback cycle.

### 6.6.1 Data Representation of the Assignment Management

An important point of the functional requirements is to offer working spaces in the Wiki to user groups where only the assigned group member can view, edit and create pages. This can be achieved with the permission and namespace system of ScrewTurn. Namespaces can contain a set of Wiki pages and are similar to namespaces in programming languages, so page names only need to be unique within a single namespace. With the help of ScrewTurn's permission system it is possible to assign users or user groups to namespaces, as visualized in Figure 13. Usually students are arranged to student user groups and then assigned to their corresponding namespace and teachers and tutors are part of a special user group and then assigned to the namespaces they have to assess.

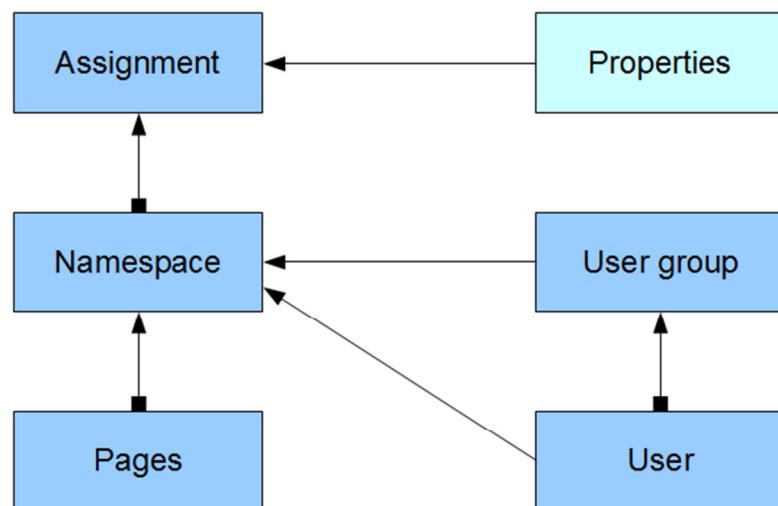


Figure 13: Hierarchical data representation of the Wiki

The hierarchical data presentation in Figure 13 above also shows that these namespaces can be arranged to collaborative writing assignments. Such assignments have properties such as the assessment scheduling and the assessment mode that are valid to all assigned namespaces.

es. Furthermore the concept of assignments needs to be implemented to provide group assessments where students need to view and assess the work of other groups (see next subsection).

## 6.6.2 Assessment and Feedback Cycle

Given that the integration of flexible assessment methods in Wikis is the central part of this thesis, it is important to conceptually design the whole assessment and feedback cycle. The assessment process has been designed as a cycle because formative assessment is very important for students, especially in computer-supported, collaborative writing assignments. Furthermore the Wiki uses rubrics as a powerful assessment tool for self and peer assessment. Thus, the remainder of this subsection explains the design of the two implemented assessment activities, the group-intern and the group-extern ones.

### 6.6.2.1 Group-intern Assessment and Feedback Activities

The group-intern assessment activities are used to provide valuable formative assessment and feedback. This is achieved with self assessment and group-intern peer assessment, which means that only group members and assigned teachers and tutors can join this feedback circle. Table 6 shows the typical workflow of this internal assessment and feedback cycle.

Participant	Action
Student 1	Checks the assignment homepage for recent changes and peer assessments and chooses a page to edit based on this information
Student 1	Diff page is shown which shows what exactly has been done by the group member (e.g. text has been added or removed)
Student 1	Peer assesses this change with a mandatory importance rating and an optional comment. This step depends on the assignment properties
Student 1	Edits the page using the Wiki WYSIWYG editor
Student 1	Self assess his own change to the page with a mandatory importance rating and an optional comment
Student 2	Repeats the same steps starting from the assignment homepage

Table 5: Group-intern assessment and feedback activities

It is also possible that the teacher or tutor adds peer assessments to the last activities of the pages in order to give formative feedback. This feedback will also be shown on the assignment homepage, which is further described in section 7.2.

### 6.6.2.2 Group-extern Assessment and Feedback Activities

For the purpose of providing summative assessment for the students, group-extern assessment activities have to be used. Here the students can group assess the work of other groups and, if configured properly, also their own work using assessment rubrics. This can not only be used to support the teacher in summative assessment and grading activities, but also to provide formative assessments for students to improve their work as described in Table 6.

Participant	Action
Group1 student	Checks and navigates through the work of the other groups and the actual assignment
Group1 student	Group assess the work and contribution of the other groups based on the whole product or a single page using an assessment rubric
Group2 student	Checks the group assessments and suggestions of the other groups, teachers and tutors
Group2 student	Improves his work based on the group assessments and suggestions
Group1 student	Checks the improvements of the other groups
Group1 student	Edits the given group assessments based on the improvements of the other groups using the assessment rubric

Table 6: Group-extern assessment and feedback activities

The teacher uses the same rubric for providing group assessments, but has a more detailed view on the group products. This includes visualizations to identify the individual contributions as further illustrated in section 7.3.

## 6.7 Summary

The aim of the practical part of this thesis is to present and implement solution approaches for these identified issues of combining CSCL with E-Assessment. Thus, a Wiki system has been enhanced with assessment methods and other tools to support teachers and students in collaborative writing activities. This practical part has been developed as part of the ALICE project, which stands for *Adaptive Learning via Intuitive/Interactive, Collaborative and Emotional Systems*. ALICE is a project in the ICT (*Internet and Communication Technologies*) area and is co-funded by the *European Commission under the Seventh Framework Programme for Research and Technological Development* (FP7). The purpose of this prototype is to enhance an existing Wiki system with flexible assessment methods to increase the engagements of students and learners. This means that the system should not only be usable like a typical Wiki system with all its advantages, but also be enhanced with integrated as-

## Chapter 6: Requirements and Design of the Prototype

essment features and other motivational aspects. These enhancements should support students and teachers in collaborative writing assignments.

The requirements are divided in two parts, one describing the functional requirements and the other describing the non-functional requirements. The functional requirements deal with the assignment authoring (user accounts, user groups, namespaces and assignment), the assessment activities (internal peer assessment, self assessment, group assessment and tagged teacher feedback), the teacher features (assignment overview and grading schema), the provided student views (typical Wiki functionalities, assignment homepage and student specific assessment features) and the integration requirements. On the other hand, the non-functional requirements explain technical requirements such as the programming style, psychological requirements such as the planned experimentations, general project requirements and the psychological requirements like experimentations.

One of the most important decisions in the design phase was to choose an appropriate Wiki system that can be enhanced in order to fulfil the requirements. This Wiki software should be free, open-source, based on the Microsoft Asp.NET framework, easily extendible, used often and similar to MediaWiki. On the basis of these limitations three Wikis (FlexWiki, MindTouch DekiWiki and ScrewTurn Wiki) have been checked and ScrewTurn Wiki has been chosen because of its flexible architecture and its big user community. The architecture of ScrewTurn consists of several components: the Web application, the core, the ACL engine, the search engine, the plugin framework and the different plugins. The flexible provider system of ScrewTurn makes it very easy to increase its functionality via plugins. These plugins can even be installed in the Wiki at runtime.

Another important design decision is how collaborative writing assignments get authored and managed in the Wiki. As shown in Figure 13, namespaces can be part of assignments and student groups and teachers/tutors can be assigned to these namespaces. Regarding the assessment and feedback cycle, it is distinguished between group-intern and group-extern activities (see Table 5 and Table 6). Group-intern assessment is used for peer assessing the page changes of other group members and for self assessing own changes. On the other hand, group-extern assessment is used for group assessing the work products of other groups and also acts as the assessment instrument of the teachers and tutors. The next chapter describes the development of the Wiki prototype based on the requirements list and the design that have been explained in this chapter.

## 7 Development of the Prototype

The last chapter has given an overview of the project idea, the requirements, the Wiki framework and the design of the prototype. On the basis of these design decisions, the aim of this chapter is to describe the development process of this project. Thus, first the technologies, frameworks and tools are listed that are needed to fulfil the required implementation tasks. Afterwards, the whole implementation process of the Wiki functions is illustrated. This is done from the developer's point of view, whereas the user's point of view is illustrated in the next chapter. Furthermore, the most important aspects of the Wiki integration are specified.

### 7.1 Used Technologies, Frameworks and Tools

This section gives a summary of the used technologies, frameworks and tools that are used to realize this project. They have been chosen based on the previous experience of the project team and on the requirements. One of the main technical requirements is to develop the Wiki based on the Microsoft .NET framework which also limited the range of choices.

#### 7.1.1 Technologies

This subsection sums up the used technologies, programming languages, protocols and techniques that were used to develop the Wiki.

##### MySQL

MySQL (Structured Query Language) is an open-source relational database management system which is used as the data backend of this system. The databases can be easily maintained with graphical user frontends such as the MySQL Administrator, the Query-Browser and the Workbench. The MySQL Connector.NET library implements the necessary Ado.NET interface to get access to MySQL databases within a .NET language such as C# or VB.NET. (MySQL, 2012)

##### C#

C# is an object-oriented programming language which can be used with Microsoft .NET. The syntax and semantic can be seen as a mixture of 70% JAVA, 20% Visual Basic and 10% C++. This means that it supports not only an automatic garbage collection and safe

## Chapter 7: Development of the Prototype

object access through references, but also the use of pointers within so called unsafe code blocks. (MSDN, 2012)

### JavaScript

JavaScript is a client side scripting language that is implemented as part of a Web browser and can be embedded in HTML (Hypertext Markup Language) pages. JavaScript and the Document Object Model (DOM) were used on different pages in this project to increase the performance and interactivity of the Web controls. This project also uses JQuery<sup>30</sup> in order to simplify the access to DOM elements and to ensure cross-browser compatibility. (w3schools, 2012)

### ViewState

ViewState is the name of the page-level state mechanism of Asp.NET. Because of the fact that the Web is stateless, the programmer has to manually save the state of a page into the so called ViewState collection across post-backs. Asp.NET offers the possibility to automatically save serialized objects into hidden fields of a Web page and unserialize them when they are needed. There also exist the session state, which can hold objects across a complete user session, and the application state, that exist for the whole lifetime of the Web application and is shared among all user sessions. (MSDN, 2012)

### AJAX

AJAX stands for Asynchronous JavaScript and XML (Extensible Markup Language) and is a technique to create highly responsible Web pages. In AJAX application, JavaScript is used to asynchronously call a Web service and to use the returned XML representation to update the page. This has the big advantage that only parts of the page can be updated and so hides the server communication from the user. (w3schools, 2012)

### SOAP

The Simple Object Access Protocol is one way to exchange information in service oriented system architectures, which are built on the use of Web services, to support distribution. It uses HTTP (Hypertext Transfer Protocol) for the information transfer and XML as its message format which makes it available to almost any system but in some case also slower than other technologies. (w3schools, 2012)

---

<sup>30</sup> <http://jquery.com/>; visited on 06<sup>th</sup> of March 2012

### 7.1.2 Frameworks

This subsection lists the frameworks that set the basis for the implementation of the Wiki. These are programming frameworks and frameworks that are needed for deploying and testing the application.

#### Microsoft .NET

The Microsoft .NET technology is used to execute programs written in .NET languages. Like JAVA, it offers a virtual machine for the execution of the generated intermediate code, but in the case of .NET this is not limited to one programming language. This opens the opportunity to the developer to mix different languages (e.g. C#, Visual Basic.NET, J#) in his or her programs. (MSDN, 2012)

#### Asp.NET

Asp.NET is Web application framework which is used to build dynamic Web sites using .NET languages. It offers a modern code-behind model to separate the presentation and content of pages. Another important feature of Asp.NET is the concept of user controls which encapsulate parts of Web pages as a single control. This offers the possibility to reuse these controls on different pages which highly increases the maintainability. (Microsoft, 2012)

#### Asp.NET Web Forms

The Web Forms framework is one way to create Web applications together with Asp.NET. In contrast to the Asp.NET MVC framework, it tries to make the development of Web applications as similar as possible to the development of Win Forms desktop applications. (Microsoft, 2012)

#### Asp.NET Web Services

Asp.NET Web services offer a very easy way to implement SOAP based Web services with Microsoft .NET. The framework fully implements the whole infrastructure for the programmer which means that he or she only needs to define and deploy his functions to make them available to anyone who need them. (Microsoft, 2012)

#### Microsoft Internet Information Services

The Microsoft Internet Information Services (IIS) is a Web server platform to deploy Asp.NET Web application or services. There also exists plug-ins to deploy PHP or JSP

## Chapter 7: Development of the Prototype

Web applications with the IIS. To run the system on Linux machines the open-source framework Mono<sup>31</sup> can be used. (MSDN, 2012)

### NUnit

NUnit is a unit-testing framework for .NET application is based on JUnit<sup>32</sup>. It can be directly integrated into Visual Studio and is used for testing classes and Web services. (NUnit, 2007)

### 7.1.3 Tools

This subsection gives an overview of the used tools that supported the development process. These are not only internal tools like programming libraries, but also external ones like versioning tools.

#### Microsoft Visual Studio 2008

The Microsoft Visual Studio is an integrated development environment (IDE) to create .NET based applications. It offers a powerful debugger to debug programs during runtime. If no full version of the Visual Studio is available, also the free Visual Web Developer can be used to develop and build Web applications. For Linux machines there exists the Mono Develop IDE. (Microsoft, 2012)

#### Asp.NET AJAX Library

The Asp.NET AJAX library contains a set of controls to build AJAX enabled Web applications with Asp.NET. Furthermore it provides the so called Update Panel that simplifies the use of AJAX without the need to code JavaScript because Postbacks that happen inside an Update Panel container only update the parts of the Web page that are inside the same container. Another AJAX library which is used in this project is the one from Anthem<sup>33</sup> that works in a very similar way but offers other controls. (Microsoft, 2012)

#### Asp.NET Charting Control

The Asp.NET charting control toolbox is a free library from Microsoft that can be used to enhance Web pages with charts and diagrams. The charts can be visualized as different types such as bar or column charts, with or without AJAX interactivity and in 2D or 3D. (Microsoft, 2012)

#### NDoc

NDoc is a code documentation generation tool for Microsoft .NET based languages to create MSDN-style documentation Web sites. For the generation, it uses the XML-style

---

<sup>31</sup> <http://www.mono-project.com/>; visited on 05<sup>th</sup> of March 2012

<sup>32</sup> <http://www.junit.org/>; visited on 06<sup>th</sup> of March 2012

<sup>33</sup> <http://anthem-dot-net.sourceforge.net/>; visited on 08<sup>th</sup> of March 2012

## Chapter 7: Development of the Prototype

code documentation of method signatures, classes, enumerations and structures. (NDoc, 2005)

### Firebug

Firebug is an add-on for Mozilla Firefox to debug JavaScript code while browsing Web sites. In this project it was used to debug client-side executions and to check DOM structures. (w3schools, 2012)

### Assembla SVN Workspace

For having a software versioning and revision control system, a free Apache Subversion repository of Assembla is used. The SVN workspace of TU Graz cannot be used because it can only be used by members of TU Graz and the project repository has also to be accessed by developers of IWT. For accessing the repository directly from the Visual Studio explorer, AnkhSVN<sup>34</sup> can be used. (Assembla, 2012)

The next section explains the implementation process of the prototype with its most important facts.

## 7.2 Implementation of the Standalone Prototype

The last section described the main programming aspects of ScrewTurn Wiki to give an overview of the general system design. The aim of this section is now to explain how the new Wiki features have been implemented based on the ScrewTurn Wiki architecture in order to fulfil the requirements. Thus, first the system overview and the database model are illustrated and a summary of Wiki features from the developer's point of view is given. After this, the most important IWT integration aspects are described.

### 7.2.1 System Overview

The system overview diagram in Figure 14 is based on the ScrewTurn architecture diagram presented in Figure 11 and shows how the different components work together. Dark components have been extended as part of this project, bright components are standard ScrewTurn ones and empty components are external ones that communicate with the Wiki.

---

<sup>34</sup> <http://ankhsvn.open.collab.net/>; visited on 08<sup>th</sup> of March 2012

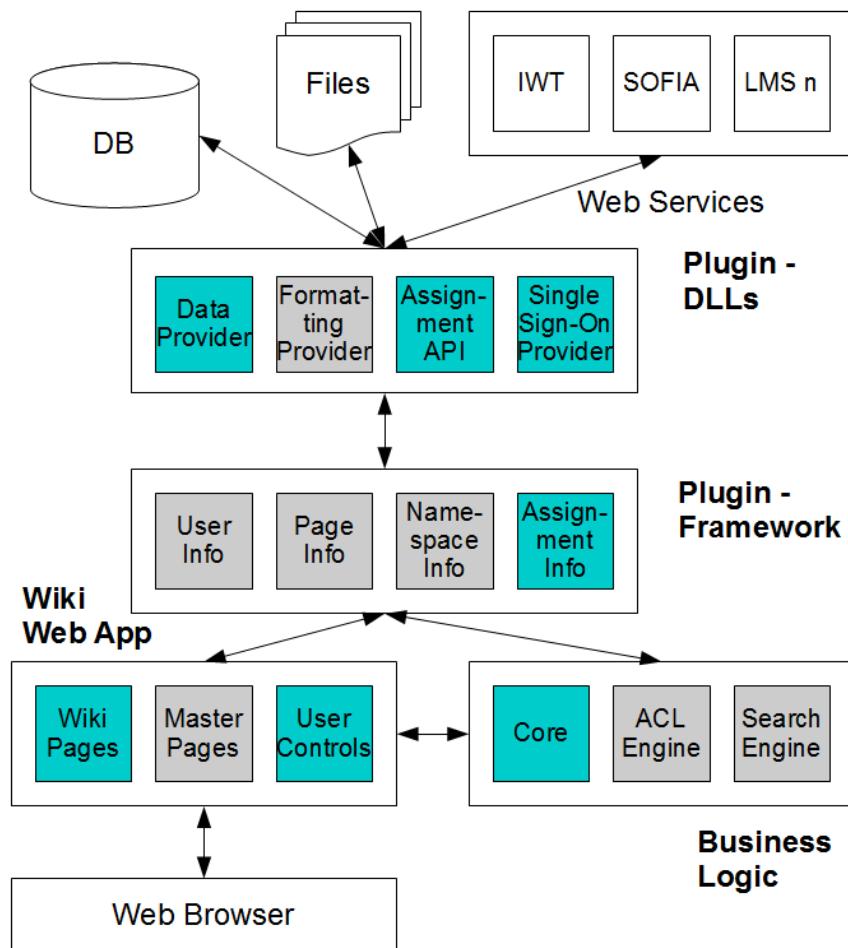


Figure 14: System overview of the standalone prototype

It has been one design principle to implement most of the new functionalities and data structures into plugins to keep the original system architecture of ScrewTurn Wiki as it is. This has been especially achieved for all the integration aspects that are needed to communicate with LMSs via Web services. The next subsection describes the data model of the Wiki that is implemented with the help of flexible data provider plugins.

### 7.2.2 Data Model

This subsection gives a summary of the database model of the Wiki. It has to be mentioned that only the new tables, which have been added to the already existing ScrewTurn database model, are described in detail. The tables that were added for this project are marked with the prefix "*alice\_*". To enhance readability, the remainder of this subsection is divided into two parts: the assessment side of the data model and the assignment side of it. The assessment side shows all tables that are needed to implement the required assessment functionalities of the Wiki and is visualized in Figure 15.

## Chapter 7: Development of the Prototype

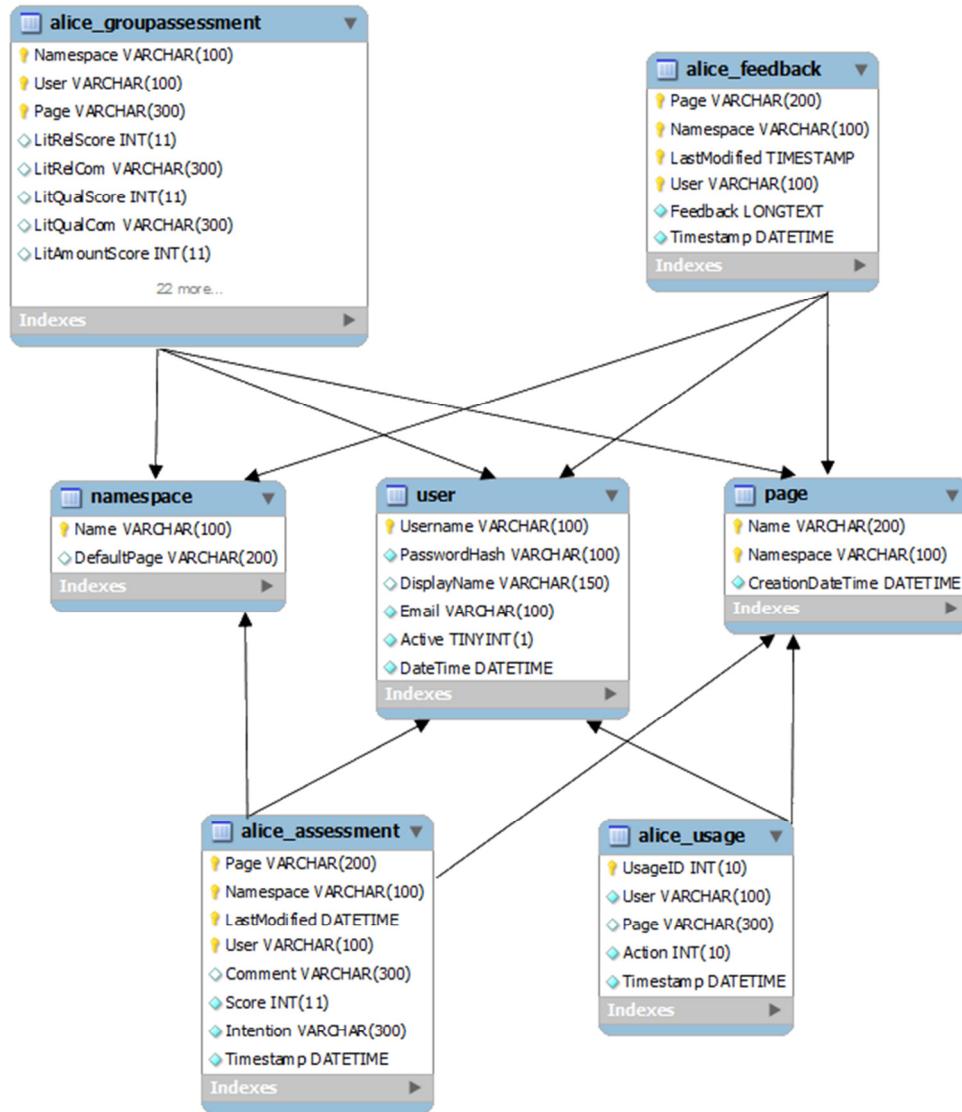


Figure 15: Assessment side of the database model

Figure 15 shows how the assessment related database tables are connected. The three tables in the middle (*namespace*, *user* and *page*) are already provided by the ScrewTurn system and the project related tables are further described in the following listing:

- *alice\_groupassessment*: represents a group assessment object that is created using a rubric for a whole namespace or a single page by an user. For this prototype it is only designed for a static rubric with 12 fields but will be changed to a dynamic one as future work.
- *alice\_feedback*: saves the inline feedback that can be done by a teacher for a specific page revision (is defined by the page, the namespace and the date of the last modification).

## Chapter 7: Development of the Prototype

- *alice\_assessment*: represents the self or peer assessment done by a user for a specific page revision.
- *alice\_usage*: used to log assessment specific actions like the number of given peer assessments.

On the other hand, there is the assignment based side of the database model. It contains all tables are necessary to provide collaborative writing assignments and is shown in Figure 16.

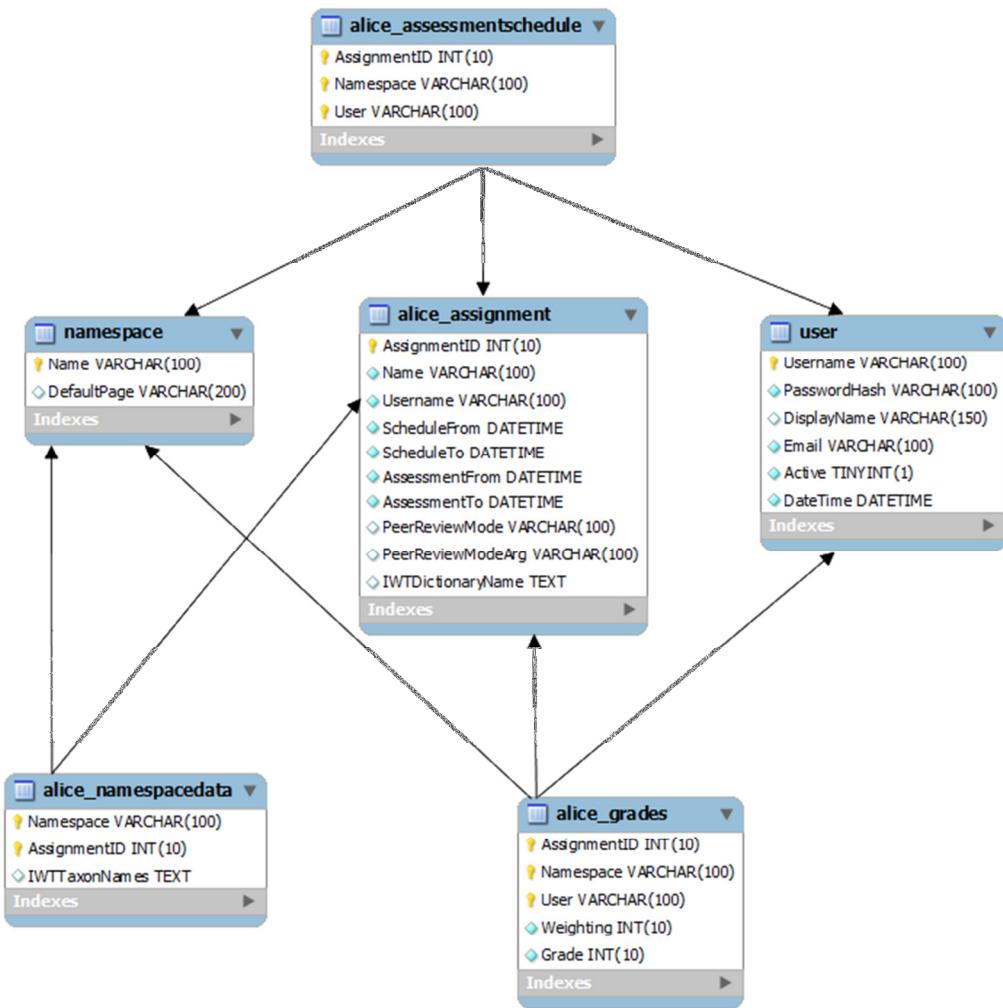


Figure 16: Assignment side of the database model

The following listing explains the assignment related tables of Figure 16:

- *alice\_assignment*: represents an assignment with all of its metadata.
- *alice\_namespacedata*: is used as a cross-table to link namespaces with assignments. Furthermore IWT specific metadata is saved here that should not be part of the ScrewTurn namespace table.

## Chapter 7: Development of the Prototype

- *alice\_grades*: saves a summative grade for a specific user of a namespace.
- *alice\_assessmentschedule*: is used to assign group assessment tasks between users and namespaces. Currently these values are directly set in the database if necessary, but there is a GUI planned for future work.

The next subsection gives an overview of the Wiki functionalities from the developer's point of view.

### 7.2.3 Programming Documentation

The aim of this subsection is to explain the most important implementation aspects of the prototype. This includes only new developments because the main ScrewTurn Wiki programming design has been already discussed in section 6.5. The Wiki functionalities from the user's point of view are explained in section 8.2.

#### 7.2.3.1 Data Repository

The data handling of the assignment and assessment specific classes is done in an own assembly (*AlicePlugin*) to separate the data access from the business logic. This goal is supported by using the *repository pattern* which is shown in Figure 17.

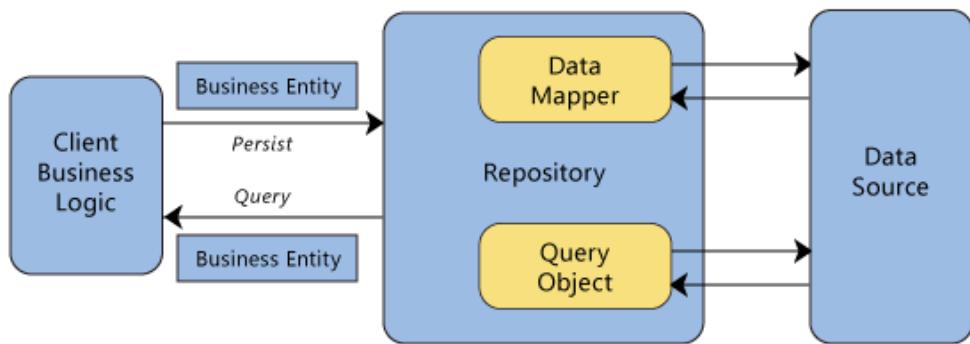


Figure 17: Interactions of the repository pattern (MSDN, 2012)

The aim of the repository pattern is to separate the logic of the data object from the client business logic for avoiding duplicated code, simplifying unit testing and providing an easy way to change the data source (e.g. databases, Web services or simple files). To do this, the repository queries the data source and maps the retrieved data on business entity objects that are exchanged with the business logic. This has the big advantage that the business logic classes only need to know the interface of the repository and the structure of these entity objects. (MSDN, 2012)

For the purpose of this project the *IAliceDataProvider* interface has been defined that contains all methods to persist and query the project specific data such as assignments, assessments, assessment schedules, group assessments, teacher feedbacks, namespaces associations, final grades and usage patterns. This interface gets implemented by *SqlServerAliceDataProvider* class that can be used to query Microsoft SQL Server and MySQL databases. The instance of this class is accessed using the static *ProviderAccess* class and the *Singleton* pattern.

### 7.2.3.2 Text Difference Algorithm

The text difference algorithm used in this Wiki is based on the work of Myers (1986) and the C# implementation of it by Matthias Hertel<sup>35</sup>. It is used to find the differences between two page revisions in order to implement the visualizations that can be seen on the diff page and the teacher's page shown in subsection 8.2.4. The original algorithm computes unique numbers for all text lines because comparing numbers is much easier than comparing text lines. Then it uses the longest common subsequence (LCS) and the shortest middle snake (SMS) methods to create an array of objects that identify deleted and inserted lines. (Myers, 1986)

For the sake of this project, the algorithm has been changed to fulfil the requirements. The following listing shows these changes and enhancements:

- The algorithm now works on the level of words and not only text lines to let the users exactly know what has happened between two revisions.
- Edit actions are tracked and visualized. An edit action can be identified if an insert action directly follows a delete action at the same position.
- Style change actions are recognized. A style change is an edit action where only Wiki markup has been added or removed before and after the word.
- All the actions done to a revision get tracked by the algorithm to list them for each page revision in the action feeds on the assignment home page.
- Mark all words added by a student to a page by iteratively calling the algorithm for all revisions and tracking the inserts. This feature can be used to visualize the individual contribution of a student on a specific page like it is done on the different charts that are used in the Wiki.

---

<sup>35</sup> <http://www.mathertel.de/Diff/DiffDoku.aspx>; visited on 15<sup>th</sup> of March 2012

## Chapter 7: Development of the Prototype

Especially the last point is very important to support the teacher in assessing the individual contributions of the students to tackle the free-riding problem mentioned in subsection 5.2.1. To use the text difference functionality the static methods in the *DiffTools* class can be used. Furthermore these methods are wrapped in the *PageContribution* class to directly receive the colour coded HTML string of a Wiki page that shows which parts have been created by which student. Figure 18 shows an example for this where one student has provided the coloured content and another student the transparent content.

The mainstream view engine used in the ASP.NET MVC Framework is the Web Forms view engine, which ships with the framework itself but is not very HTML friendly. By default, the view engine in the MVC framework uses regular .aspx pages to design the layout of the user interface pages onto which the data is composed. However, different view engines can be used. Additionally, rather than the default ASP.NET postback model, any interactions are routed to the controllers using the ASP.NET Routing mechanism. Views can be mapped to REST-friendly URLs.

Figure 18: Example of the Wiki contribution visualization

### 7.2.3.3 Assessment User Controls

In Asp.NET a user control is a container for other user controls that is needed multiple times in a Web application to ensure maintainability. Such a user control has the same lifecycle as an Asp.NET Web page and can have properties and class methods that can be used for data binding or other operations. In the course of this project, the assessment and feedback functionalities for self assessment, peer assessment and group assessment have been implemented using the principle of user controls.

The main assessment control is called *AliceRatingAssessment* and is visualized in Figure 19. It can be configured to be read-only for feedback purposes and to hide the intention checkboxes for peer assessment purposes. The importance rating is implemented using the Rating control of the AJAX toolkit.

The screenshot shows a user control interface for self-assessment. At the top, there is a label 'Intention: \*' followed by a list of checkboxes:

- Added new paragraph
- Edited paragraph
- Changed style
- Created new page
- Provided comment
- Other

Below this is a label 'Comment:' followed by a text input field containing the text 'Added explanations'. A green rectangular border highlights this input field. At the bottom, there is a label 'Importance:' followed by a star rating control showing four yellow stars.

Figure 19: Assessment control used for self assessment

In order to increase maintainability and the look-and-feel consistency, the assessment control above has been reused to implement the group assessment rubric. In the current ver-

sion of the prototype, a rubric is defined by twelve criteria and so also twelve controls were used for the rubric. Furthermore the assessment control has been enhanced with properties for the title, the weighting and a tooltip text. For feedback purposes, it is also possible to directly set all the fields based on a *GroupAssessment* object.

#### 7.2.3.4 Graph Controls

One important requirement and goal of this project is the implementation of visualization aspects. Microsoft offers a free library for Asp.NET compatible graphs and charts to realize visualization functionalities as presented in subsection 5.2.1. This library has been wrapped into user controls that provide user friendly interfaces to populate the charts with the Wiki data. These user controls are named *AliceRatioChart* for multiple series used by the teacher page and *AliceRatioChartSingle* for a single series used by the student pages. An example for the second one, that visualizes the student contribution based on the number of provided letters, can be seen in Figure 20.

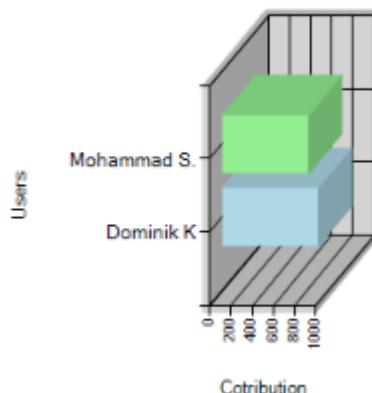


Figure 20: Example of the graph control to visualize student contributions

Moreover, the Asp.NET graph library supports AJAX functionality when clicking on a chart library. To use this in a flexible way, the interfaces of the wrapper controls can register an event handler. This has the big advantage that each page can define the behaviour of the click events of its chart (e.g. show the content of the clicked page revision).

#### 7.2.3.5 WYSIWYG Editor

A Wiki provides a way to easily create a Web page without the need of knowing HTML using a simple Wiki markup language. Although a lot of users prefer a Microsoft Word like WYSIWYG (What you see is what you get) editor. ScrewTurn Wiki uses such an editor to format the content of a Wiki page and to add links to other Wiki pages.

## Chapter 7: Development of the Prototype

The editor control is designed in a very flexible way and so it is easy to add new tabs and buttons to it. This means that the editor can also be used to implement the inline feedback functionality which is used by the teacher to mark specific parts of a page with colour codes. The meanings of these colour codes can be seen in Table 7.

Colour Code	Meaning
	Check spelling
	Check style
	Extend text
	Shorten text
	Rewrite text
	Add/check reference
	Add/check link

Table 7: Inline feedback colour codes

Within the editor this new functionality is implemented using the already provided *WrapSelectedMarkup* JavaScript function to add a HTML span with the specific background colour (e.g. the RGB (red green blue) code #F5A9A9 to indicate that the spelling should be checked).

### 7.2.3.6 Web Application

The Web application is the visual part of Wiki that interacts with the users. It contains the Asp.NET Web pages that are provided for the users via the HTTP protocol. In order to guarantee uniform layout and design, the Wiki uses two Asp.Net master pages, one for the standard Wiki pages and one for the administrator area. These master pages also handle the visibility of the shown controls based on the permissions of the currently logged in student. The following listing describes all pages that were added to ScrewTurn Wiki as part of this project:

**AliceAdminAssignments:** This page is used to view, create, edit and delete assignments. It uses the repeater user control to list the assignments with all possible actions that can be done to them. This repeater control can be directly bound to a list of assignment business objects using the powerful data binding feature of Asp.NET. Moreover, the Anthem AJAX calendar control is used on this page to choose the group assessment schedule.

**AliceContribution:** The students use this page in the group assessment phase of the assignment. The assessment rubric is shown in a popup that is implemented using the *ModalPopupExtender* of the Microsoft AJAX toolkit. Furthermore, the Asp.NET ViewState concept is used to implement the page navigation player.

## Chapter 7: Development of the Prototype

**AliceDiff:** Shows the text differences between two page revisions.

**AliceEdit:** Is used to change the content of a Wiki page and uses Asp.NET validation controls for the client-side validation of input controls.

**AliceFinalGrades:** This page can be used by the teacher to give summative grades. It is enhanced with controls of the AJAX toolkit such as the *NumericUpDownExtender* and the *TreeView*.

**AliceMotivation:** Shows the motivate/progress charts. This page uses LINQ (Language Integrated Query) and Lambda expressions for querying the student contributions in the specific assignment phases.

**AliceNamespaceActions:** The action feeds and revision histories are visualized on this page. Moreover, it shows social aspects like the online user list. To update this list without updating the whole page, the AJAX update panel is used.

**AlicePageGraph:** This page shows the link structure of the pages of a specific namespace and has been implemented by another member of the project team and not by the author of this thesis. It uses the Birdeye visualization library<sup>36</sup> which is written in Flash.

**AliceTeachersView:** Is used by the teacher to check and to assess the students' individual contributions.

**Global.asax:** The global.asax file is not a Web page; in fact it represents the HTTP Web application with its Web based events. This includes events that get called when a user session starts and ends which is used for tracking the online user list that is shown on the assignment home page.

This subsection has shown the features of the prototype from the developer's point of view. The aim of the next subsection is to show how the Wiki can be integrated into LMSs.

## 7.3 Integration of the Prototype

As mentioned in chapter 6, it is a main requirement of the Wiki prototype that its usage is flexible. This means that it should be usable not only as a standalone Web application but also as a Web tool integrated into a Learning Management System (LMS) like Moodle. As part of this master thesis and the ALICE project, the Wiki prototype has been fully integrated with the Intelligent Web Teacher (IWT) LMS as a usage scenario. Furthermore the

---

<sup>36</sup> <http://code.google.com/p/birdeye/wiki/RaVis>;

## Chapter 7: Development of the Prototype

Wiki also has been integrated with the SOFIA (Service-oriented Flexible and Interoperable Assessment) system to test the SSO mechanism (AL-Smadi & Guetl, 2011). However, this section also shows how the Wiki can be integrated into other LMSs.

The remainder of this section is divided into three parts. First the login process via Single Sign-on (SSO), which is the core component of this integration plan, is described. After this, the assignment authoring Web service interface and the grading schema page, that can be used to update learner models based on the Wiki contributions, are illustrated. The sequence diagrams used in this section have been created in collaboration with other members of the ALICE project staff to agree on the programming interfaces and will also be published in edited versions in official ALICE deliverables.

### 7.3.1 Login via SSO

The main part of the Wiki integration is to make it accessible and useable via other LMSs. This means that a LMS could have multiple student tasks which are provided as a learning sequence and one of these tasks could be a collaborative writing assignment using the Wiki prototype. In this case a user should be able to directly access the Wiki without the need of doing a second login or even registration. In order to achieve this goal, a Single Sign-on (SSO) approach is used. SSO is defined as an authentication mechanism that enables a user to login once to a Web application and to gain access to further services and tools, which are integrated with this application, without the need to login several times. This not only reduces time, but also increases security because the user only has to remember one username/password combination. (Wang, Chen, & Wang, 2012)

To design this approach in a flexible way, it is important that the whole user data is only saved in the database of the LMS and not in the LMS and the Wiki, which would highly increase the maintenance effort. Thus, a user provider (see subsection 6.5.2) has to be implemented that fetches the necessary user data from the corresponding LMS using Web services. This also increases the privacy and security because the Wiki can only access the user data that is available via the installed providers. For this project, a user provider has been implemented that can access data from the IWT portal, but new providers for other portals can easily be implemented and registered using the flexible plugin system of ScrewTurn Wiki. The whole sequence of this authentication process is illustrated in Figure 21.

Here a user wants to access the Wiki via a Web application which opens a special *SSOLogin* page in the Wiki. This page just redirects the request with all necessary parameters like the corresponding username, the role and the user group to the auto login mechanism of the Wiki. This mechanism uses the implemented user provider to fetch the user

## Chapter 7: Development of the Prototype

data from the calling Web application which also logs the user into the Wiki. It is important to say that all further accesses to the user data is also done via this provider because the Wiki should only save Wiki related data such as namespaces, pages and permissions and no user related data such as accounts and user groups. Another important thing happens on the *SSOLogin* page in step 2 where the *Authenticate* method is called. This method checks if the Web application that accesses the Wiki is allowed to do so using the *OAuth* standard. OAuth is a way to achieve delegated authorizations. This means that a service provider allows access to a consumer using access tokens or shared secrets. (Gonzalez, Rodriguez, Nistal, & Rifon, 2009)

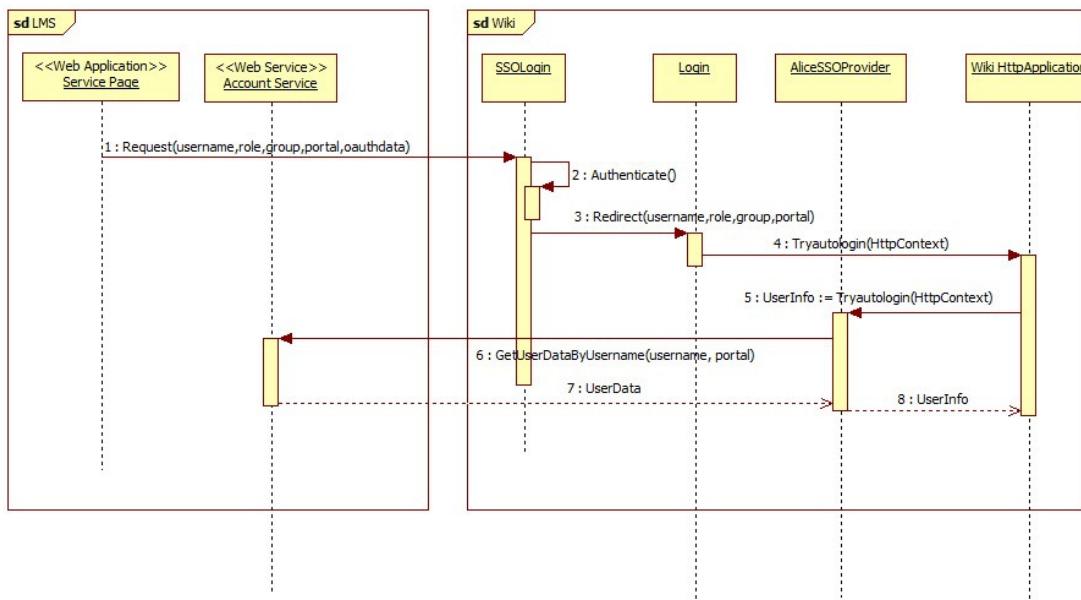


Figure 21: Sequence diagram of the SSO login process

Within this project the service provider is the Wiki and the consumer is the IWT portal and both applications share a secret that is used for the authorization process. This is developed using an open source implementation of the OAuth protocol<sup>37</sup> which checks if the calculated hash value based on the shared secret is correct. This works because only authorized application which know the shared secret can calculate this value. The next subsection gives an overview of the implemented Web service interface that is used for authoring collaborative writing assignment. Furthermore it describes how user groups of the LMS can be mapped to user groups and namespaces of the Wiki.

<sup>37</sup> <http://oauth.googlecode.com/svn/code/csharp/>; visited on 04<sup>th</sup> of April 2012

### 7.3.2 Assignment Authoring Web Services

The last subsection has explained how the Wiki can be accessed from LMSs via SSO, but it is also important that the Wiki provides assignments and namespaces for the users that want to access it. Thus, it is necessary that the teachers can create this data within the LMS based on the requirements of their courses. In order to achieve this goal, a Web service interface has been implemented and published that offers CRUD operations for assignments, namespaces and ACL permissions. To ensure consistency, it is essential that user group names in the LMS are directly mapped to namespace names in the Wiki and that teachers are part of a user group called “*Teachers*”. The sequence of this communication can be seen in Figure 22.

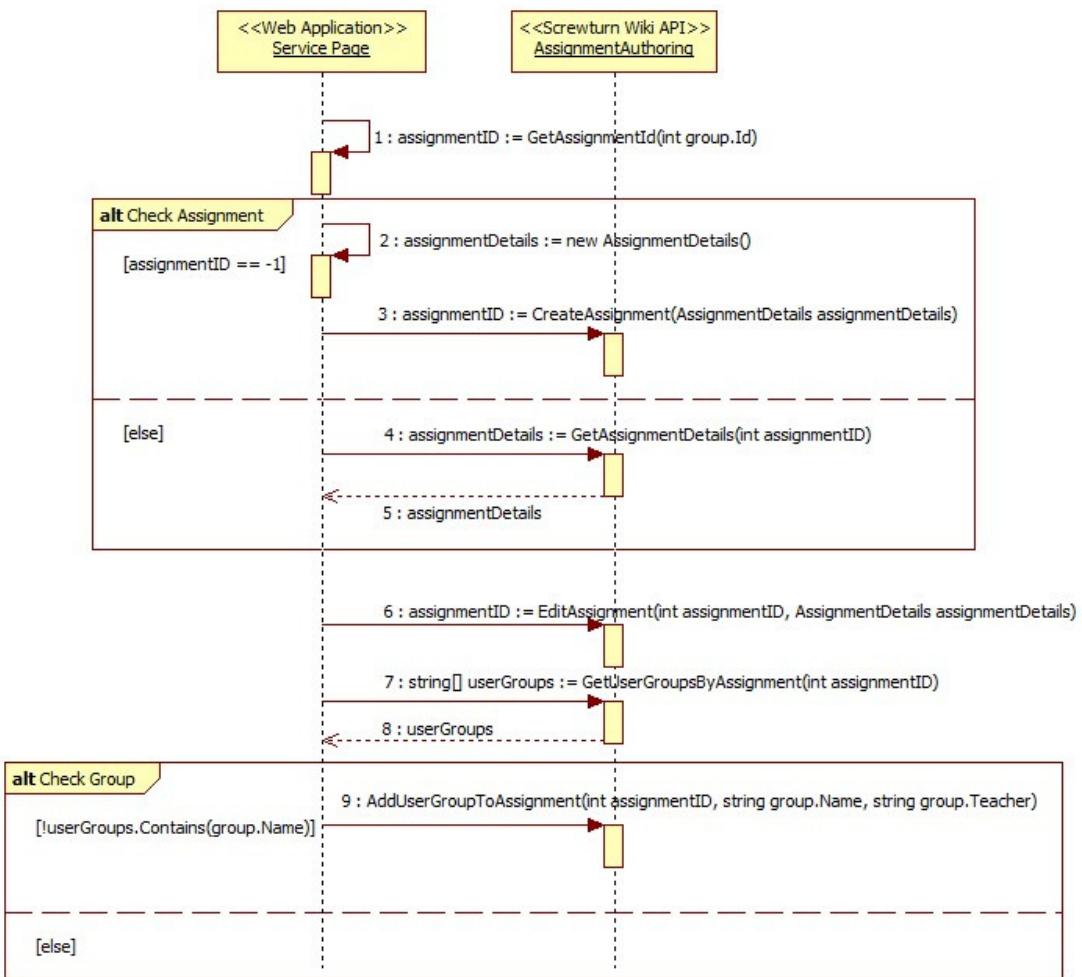


Figure 22: Sequence diagram of the assignment authoring process

The Web application first checks if an assignment for the currently selected user group exists and creates a new one if not. The *AssignmentDetails* structure holds all assignment relat-

## Chapter 7: Development of the Prototype

ed data like the scheduling dates and an optional competence dictionary which will be further explained in the next subsection. After an assignment has been created, user groups can be added to it using the *AddUserGroupToAssignment* method. This method is the most complex one because it needs to create the namespace for the corresponding group, its main page and all permissions. Furthermore, its method signature also contains the username of a teacher because it could be possible that each group should be assessed by another teacher or tutor. The Web service interface also contains edit and delete-methods for all the data that is not shown in Figure 22.

Another important thing regarding the Web service implementation is that the services get called outside the environment of the Wiki Web application. This means that all necessary providers for users, pages, files and settings need to get instantiated and registered manually at the beginning of each call. The next subsection explains the grading schema of the Wiki that is used to update the learner models of the LMSs.

### 7.3.3 Grading Schema

In the last subsection it was mentioned that the assignment can be authored with a competence dictionary. This dictionary contains competencies or concepts that should be mastered within this collaborative writing assignment (e.g. citation styles or formal writing aspects). These concepts can be assigned to the user groups in order to provide summative assessments and to update the learner models using a special grading schema Web page. The interactions of this process are visualized in Figure 23.

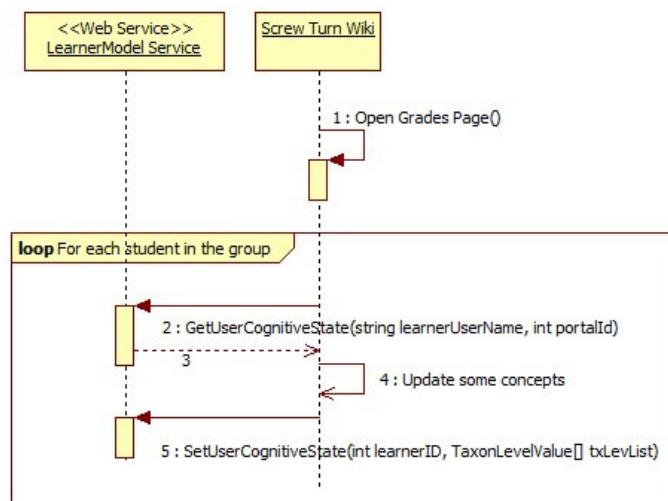


Figure 23: Sequence diagram of the grading process

In this use case for the IWT integration, a teacher opens the grading schema page for giving summative assessments based on the assigned concepts. Thus, this page interacts with two Web services to receive the current cognitive state<sup>38</sup> and to save back the updated cognitive state<sup>39</sup> of a student. For future work these concepts could be mapped to master levels of the assessment rubric that is used for group assessment. In this case the system could suggest a grading value for each concept to the teacher based on the given assessments.

## 7.4 Summary

The aim of this chapter is to describe the development process of this project from the developer's point of view. Thus, first the technologies, frameworks and tools are listed that are needed to fulfil the required implementation tasks. Afterwards, the whole implementation process of the Wiki functions is illustrated. Furthermore, the most important aspects of the Wiki integration are specified.

The technologies, frameworks and tools have been chosen based on the previous experience of the project team and on the requirements. One of the main technical requirements is to develop the Wiki based on the Microsoft .NET framework which also limited the range of choices. Thus, the Wiki uses Microsoft Asp.NET Web forms as its Web application framework and C# and JavaScript for the server-side and client-side programming. Furthermore, .NET related tools such as the Visual Studio IDE, NDoc, NUnit and AJAX user control libraries are used.

The system architecture of the standalone Wiki prototype is based on the one of ScrewTurn Wiki and contains four main components, the Wiki Web application itself, the business logic assemblies (core, ACL engine and search engine), the plugin framework middle ware and the plugins that communicate with the external components such as databases or other LMSs. The data model is divided in the assessment side and the assignment side of the system and the data access is done in an own assembly (*AlicePlugin*) to separate the data access from the business logic. This goal is supported by using the *repository design pattern*.

Other implementation aspects are the text difference algorithm, the assessment user controls and graph controls, the WYSIWYG editor and the Wiki Web application. The text difference algorithm used in this Wiki is based on the work of Myers (1986). It is used to find the differences between two page revisions in order to implement the visualizations

---

<sup>38</sup> <http://iwtalice.crmpa.unisa.it/iwt/RemoteServices/ExternalServices/KnowledgeManagerServices.asmx>; visited on 05<sup>th</sup> of April 2012

<sup>39</sup> <http://iwtalice.crmpa.unisa.it/iwt/RemoteServices/ExternalServices/LearnerModelServices.asmx>; visited on 05<sup>th</sup> of April 2012

## Chapter 7: Development of the Prototype

that can be seen on the diff page and the teacher's page shown in subsection 8.2.4. The algorithm computes unique numbers for all text lines because comparing numbers is much easier than comparing text lines. In Asp.NET a user control is a container for other user controls that is needed multiple times in a Web application to ensure maintainability. Such a user control has the same lifecycle as an Asp.NET Web page and can have properties and class methods that can be used for data binding or other operations. In the course of this project, the assessment and feedback functionalities for self assessment, peer assessment and group assessment have been implemented using the principle of user controls. Moreover, the AJAX graph library of Microsoft has been wrapped into user controls that provide user friendly interfaces to populate the charts with the Wiki data.

The Web application is the visual part of the Wiki that interacts with the users. It contains the Asp.NET Web pages that are provided for the users via the HTTP protocol. In order to guarantee uniform layout and design, the Wiki uses two Asp.Net master pages, one for the standard Wiki pages and one for the administrator area. These master pages also handle the visibility of the shown controls based on the permissions of the currently logged in student.

As mentioned in chapter 6, it is a main requirement of the Wiki prototype that its usage is flexible. This means that it should be usable not only as a standalone Web application but also as a Web tool integrated into a Learning Management System (LMS) like Moodle. As part of this master thesis and the ALICE project, the Wiki prototype has been fully integrated with the Intelligent Web Teacher (IWT) LMS as a usage scenario. Furthermore the Wiki also has been integrated with the SOFIA (Service-oriented Flexible and Interoperable Assessment) system to test the SSO mechanism (AL-Smadi & Guetl, 2011). There are three main integration aspects, the login via SSO, the assignment authoring Web services and the update of the user LMS user models via a special grading schema. The next chapter describes the functionalities of the Wiki from the user's point of view to cover all facets of this project.

## 8 Usage Point of View and Experimentations

The last chapter has given an overview of the development of the prototype and especially has shown the functionalities of the Wiki from the developer's point of view. The aim of this chapter is to illustrate these functionalities from the user's point of view. This includes installation and setup instructions and explanations of the Wiki functionalities which can also be used like a user handbook. Furthermore, the experimentations that have been administrated as part of this project are explained. This includes also the experimentation findings and how they have influenced the development process of the Wiki. Some of the screenshots that are used in this chapter have already been published in official ALICE deliverables.

### 8.1 Installation and Setup

The following steps describe how the Wiki can be installed and be started using the files on the CD (see Appendix: CD-ROM):

1. Open the Visual Studio 2008 solution file that can be found in the trunk folder and fully build it.
2. Create a new Web site in IIS (Microsoft Internet Information Services) that uses the *WebApplication* folder as its physical path.
3. Create a MySQL database called *screeturn* that can be accessed with user *root* and empty password. The connection string of the database can be changed in the *web.config* and *app.config* files of the application.
4. Run the *co\_wiki.sql* script to create the database tables.
5. Create a temporary directory for the *ChartImageHandler*. The default directory path is *D:\CHART\_TEMP*, but this can be changed in the *web.config* file.

The SQL script also creates a demo assignment called *Test* with two topics called *Test1* and *Test2*. They can be accessed with the following accounts (username/password):

- student1 / student
- student2 / student
- teacher / teacher
- admin / iwt\_co\_wiki

There also exists an installation of the Wiki on the server of the AEMT group<sup>40</sup> that can be accessed to test the Wiki.

## 8.2 Wiki Functionalities

This section shows the Wiki functionalities from the user's point of view which also can be used as a user handbook. The remainder of this section is separated into four functionality parts, the assignment authoring, the assessment and feedback activities, the student views and the teacher views. Furthermore, only newly implemented functionalities are explained here because the standard ScrewTurn functions are already described on its homepage<sup>41</sup>.

### 8.2.1 Assignment Authoring

This subsection shows how assignments can be authored by teachers and admins. Please note that the corresponding accounts, user groups, namespaces and permissions already have to be created before the assignment creation process gets started as shown in Figure 24.

The screenshot shows a web-based assignment creation interface. At the top, there is a navigation bar with tabs: Admin Home, User Groups, Accounts, Namespaces, Assignments, and a bolded 'Assignments' tab. Below the navigation bar is a header 'Assignment Details'. A 'Name' field contains the value 'Test'. Under 'Assessment From', there is a calendar grid for January 2012. The date '19' is highlighted in grey. Under 'Assessment To', there is another calendar grid for January 2012, where the date '20' is highlighted in grey. At the bottom, there is a 'Peer Review Mode' dropdown set to 'Based on Intention' with a dropdown arrow, and a 'Peer Review Argument' input field containing 'PageCreate'. There are also 'Save Assignment' and 'Cancel' buttons.

Figure 24: Assignment authoring

<sup>40</sup> <http://alice.iicm.tugraz.at/co-wiki/>; visited on 25<sup>th</sup> of April 2012

<sup>41</sup> <http://www.screwtturn.eu/Help.MainPage.ashx>; visited on 25<sup>th</sup> of April 2012

## Chapter 8: Usage Point of View and Experimentations

An assignment is defined by its name, the scheduling dates of the group assessment phase and the internal peer review mode as explained in the requirements in subsection 6.2.1.1. After an assignment was authored, it can be selected by a teacher in his or her assignment list to allocate namespaces to it as presented in Figure 25.

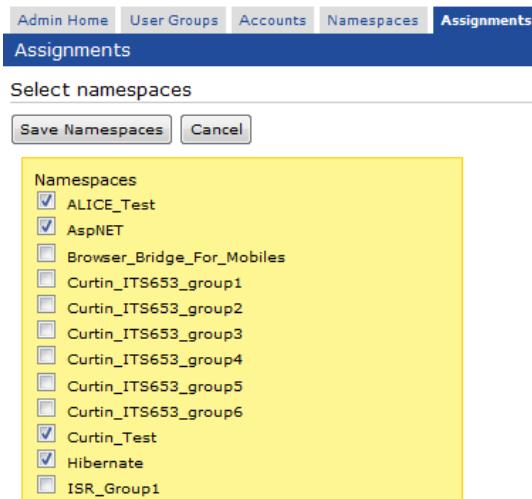


Figure 25: Allocation of namespaces to assignments

The authoring of assignments has also been a main goal in the integration process of the Wiki (see section 7.3). Figure 26 shows the IWT integration of it.

The screenshot shows an 'Assignment Authoring' form. At the top, there are buttons for 'Save' and 'Close'. The form contains several input fields: 'Name' (WebProgramming), 'Schedule Start' (01/03/2012), 'Schedule End' (01/03/2012), 'Assessment Start' (22/04/2012), 'Assessment end' (30/04/2012), 'Peer Rev. Mode' (Optional), and 'Domain' (WebProgramming). Below the form is a table with columns 'Name', 'Description', and 'Concepts'. There is one row with 'Name' MVC, 'Description' AspNet;JavaScript;JQuery;, and 'Concepts' (with edit and delete icons). At the bottom right of the form, there is a 'Zoom in' link.

Figure 26: Integrated assignment authoring

## Chapter 8: Usage Point of View and Experimentations

This page in IWT uses the implemented Web service interface to author assignments, namespaces and permissions. The accounts and user groups are managed by the LMS itself. The next subsection describes the assessment and feedback activities of the Wiki.

### 8.2.2 Assessment and Feedback Activities

This subsection shows the assessment and feedback activities that have been implemented to increase the motivation of students. First, the group-intern assessment activities are described that are used to assess page revisions created by group members. Figure 27 shows the internal peer assessment process as described in subsection 6.6.2.1.

The screenshot shows a web-based interface for comparing page revisions. At the top, there's a navigation bar with 'Compare Page Revisions' and dropdown menus for '2' and 'Current'. A 'Compare' button is also present. Below this is a color key: 'added text' (green), 'removed text' (red), 'edited text' (blue), and 'edited style' (orange). The main content area displays two revisions of a page. Revision 1 contains the text: 'The ASP.NET MVC Framework is a web application framework that implements the model-view-controller pattern. View and Controller. A model represents the state of a particular aspect of the application. Frequently, a controller handles interactions and updates the model to reflect a change in state of the application, and Importance: ★★★★☆'. Revision 2 adds 'Razor is the newest engine by MS MS. It is very good for HTML'. To the right of the text, there's a 'Peer Review' section with a 'Comment' field containing 'More explanations please' and a rating scale from 1 to 5 stars. A green 'Edit current version' button is at the bottom right.

Figure 27: Internal peer assessment on the difference page

An internal peer assessment can be given by a group member before editing a page and is based on the last revision of this page to rate the importance of it and to give helpful suggestions. On the contrary, there is the internal self assessment process shown in Figure 28. This way of assessment is used to self assess the own work when changing a page to let the group members and the teachers know about the importance and the intentions of this change.

The screenshot shows a web-based interface for editing a page titled 'MVC Components'. At the top, there are tabs for 'WikiMarkup', 'TeacherFeedback', 'Visual', and 'Preview'. Below the title is a toolbar with various icons for text formatting and media insertion. The main content area contains the page's text: 'The ASP.NET MVC Framework is a web application framework that implements the model-view-controller pattern. Based on ASP.NET, it allows SW developers to build a Web application as a composition of three roles: Model, View and Controller. A model represents the state of a particular aspect of the application. Frequently, a model maps to a database table with the entries in the table representing the state of the application. A controller handles interactions and updates the model to reflect a change in state of the application, and then passes information to the view. A view accepts necessary information from the controller and renders a user interface to display that.' Below this is another line of text: 'Razor is the newest engine by MS. It is very good for HTML'. To the right, there's a 'Self Reflection' section with a 'Comment' field containing 'Added explanations' and a 'Importance' rating of four stars. A list of checkboxes for 'Intention' includes 'Added new paragraph' (checked), 'Edited paragraph', 'Changed style', 'Created new page', 'Provided comment', and 'Other'. At the bottom, there are buttons for 'Save', 'Cancel', 'Save new Feedback', and 'Edit Feedback'.

Figure 28: Internal self assessment on the edit page

The second implemented assessment activity is the group-extern assessment using rubrics. This is used by students and teachers to assess group works with the help of a a rubric that can be seen in Figure 29.

## Chapter 8: Usage Point of View and Experimentations

**Assessment for group MVC**

<b>References</b> 8 / 10%	<b>Amount of References</b>	<b>Quality of References</b>		
Comment: <input type="text"/>	Comment: <input type="text"/> Great	Importance:		
Importance:				
<b>Content</b> 33 / 70%	<b>Abstract</b>	<b>Introduction</b>	<b>Historic Overview and Classification</b>	<b>Selected Aspects</b>
Comment: <input type="text"/>	Comment: <input type="text"/>	Comment: <input type="text"/>	Comment: <input type="text"/> Ok	Comment: <input type="text"/>
Importance:	Importance:	Importance:	Importance:	Importance:
<b>Formal Aspects</b> 14 / 20%	<b>References and Citations</b>	<b>Style</b>	<b>Structure</b>	<b>Appendix</b>
Comment: <input type="text"/>	Comment: <input type="text"/>	Comment: <input type="text"/>	Comment: <input type="text"/>	Comment: <input type="text"/>
Importance:	Importance:	Importance:	Importance:	Importance:

**Submit** **Close**

Figure 29: Rubric used for the group assessment

The results of the group assessment process can be seen on the assignment homepage in the “Group Peer Review” tab as presented in Figure 30.

**MVC - Home**

**Actions Feed**  
**Group Peer Review**

Show review for: All pages ▾

**Importance Ratings**

Reviews done by Daniel B.

<b>References</b> 4 / 10%	<b>Amount of References</b> Importance:  3	<b>Quality of References</b> Importance:  1		
<b>Content</b> 37 / 70%	<b>Abstract</b> Importance:  5	<b>Introduction</b> Importance:  4	<b>Historic Overview and Classification</b> Importance:  9	<b>Selected Aspects</b> Importance:  6
<b>Formal Aspects</b> 10 / 20%	<b>References and Citations</b> Importance:  2	<b>Style</b> Importance:  1	<b>Structure</b> Importance:  4	<b>Appendix</b> Importance:  3

**References Comments**  
**Content Comments**

Abstract	Introduction	Historic Overview and Classification	Selected Aspects	Lessons Learned	Summary and Future Work
Should be extended	Great				

**Formal-Aspects Comments**

References and Citations	Style	Structure	Appendix
Please more			

**Teacher Review**

Figure 30: Group assessment feedback on the home page

## Chapter 8: Usage Point of View and Experimentations

This screen shows the averaged group assessments done by the students. Teachers see here also the names of the students that are responsible for this feedback. The “Teacher Review” tab shows the group assessment done by the teacher as presented in Figure 31.

<b>References</b> 8 / 10%	<b>Amount of References</b> Comment: <input type="text"/> Importance: ★★★★☆ 3	<b>Quality of References</b> Comment: Great Importance: ★★★★☆ 5		
<b>Content</b> 33 / 70%	<b>Abstract</b> Comment: <input type="text"/> Importance: ★★★★☆ 1	<b>Introduction</b> Comment: <input type="text"/> Importance: ★★★★☆ 2	<b>Historic Overview and Classification</b> Comment: Ok Importance: ★★☆☆☆ 6	<b>Selected Aspects</b> Comment: <input type="text"/> Importance: ★★★☆☆ 18
<b>Formal Aspects</b> 14 / 20%	<b>References and Citations</b> Comment: <input type="text"/> Importance: ★★★★☆ 4	<b>Style</b> Comment: <input type="text"/> Importance: ★★★★☆ 4	<b>Structure</b> Comment: <input type="text"/> Importance: ★★★★☆ 4	<b>Appendix</b> Comment: <input type="text"/> Importance: ★★★☆☆ 2

Figure 31: Teacher's group assessment feedback

This screen also shows the final grade for the current assignment if there has been saved any (see subsection 8.2.4). The results of the internal peer assessments are usually shown in the “Action Feeds” section of the assignment homepage but can also be seen in the revision player of the progress charts page (see subsection 8.2.3) as illustrated in Figure 32.

Revision player	
<b>Play</b>	Revision 4 << < > >> Loop
<b>Peer Review</b>	
Comment: More explanations please Importance: ★★★★☆ 60	

The ASP.NET MVC Framework is a web application framework that implements the model-view-controller pattern. Based on ASP.NET, it allows SW developers to build a Web application as a composition of three roles: Model, View and Controller. A model represents the state of a particular aspect of the application. Frequently, a model maps to a database table with the entries in the table representing the state of the application. A controller handles interactions and updates the model to reflect a change in state of the application, and then passes information to the view. A view accepts necessary information from the controller and renders a user interface to display that.

Razor is the newest engine by MS. It is very good for HTML.

Figure 32: Internal peer feedback on the progress charts page

### 8.2.3 Student Views

The student views are special pages designed for students. However, the teacher can also view these pages to check the progress of the students. Figure 33 shows a rendered Wiki page that is integrated and opened in IWT.

The screenshot shows a web browser window with a blue header bar. The header contains the text "You are here: Home > Courses Catalogue > Information Search and Retrieval". Below the header are tabs for "Course", "Bookmarks", "Forum", and "Co-writing wiki". The main content area has a dark header with the text "Co-Writing Wiki". On the left, there is a sidebar titled "Navigation" with links to "Home", "MainPage", "Create a new page", "Progress Charts", "Namespace Structure", and "Peer Review". The main content area has a title "View Engines" and a sub-header "Modified on 2012/01/19 11:44:57 by Dominik K". It also shows the URL "4) > Main Page (MVC) > MVC Components (MVC) > View Engines (MVC)". The content itself discusses the Web Forms view engine and Spark, mentioning ASP.NET Routing and Razor. There are also "Discuss (0)", "Edit", and "Actions Feed" buttons at the top right of the content area.

Figure 33: Integrated and rendered Wiki page

In SOFIA, pages are not directly integrated in frames but are opened in a new browser tab. For doing the group-extern assessment activity, the Wiki offers a special page to the students that can be seen in Figure 34.

The screenshot shows a web page titled "Group assessment for : AspNet". The left sidebar has a tree-view control with nodes for "Explore groups" (Groups, WebForms), "Pages for WebForms", and "MainPage". The main content area has a navigation bar with "Back", "Main", and "Wiki player for group WebForms on page View-State". It includes a diagram showing a connection from " MainPage" to " ViewState". The text in the content area discusses View state management, its use for preserving form information across postbacks, and the need to be wary of storing sensitive information in it. There are "Actions" buttons for "Assess group" and "Assess page". A "Switch to full screen ..." button is at the bottom left.

Figure 34: Contribution visualization on the group assessment page

This page contains a tree-view control showing all other groups that have to be assessed by the currently logged-in user. If a group gets clicked, the main page of it is shown in the

## Chapter 8: Usage Point of View and Experimentations

centre of the page. Moreover, it is possible to navigate through the Wiki page structure of this namespace by directly following the links or by the help of the graph visualization on the left. A group assessment can be done on the level of the whole namespace or on the level of a single page by clicking the links on the right. One of the most important functionalities for students is the assignment home page that is visualized in Figure 35.

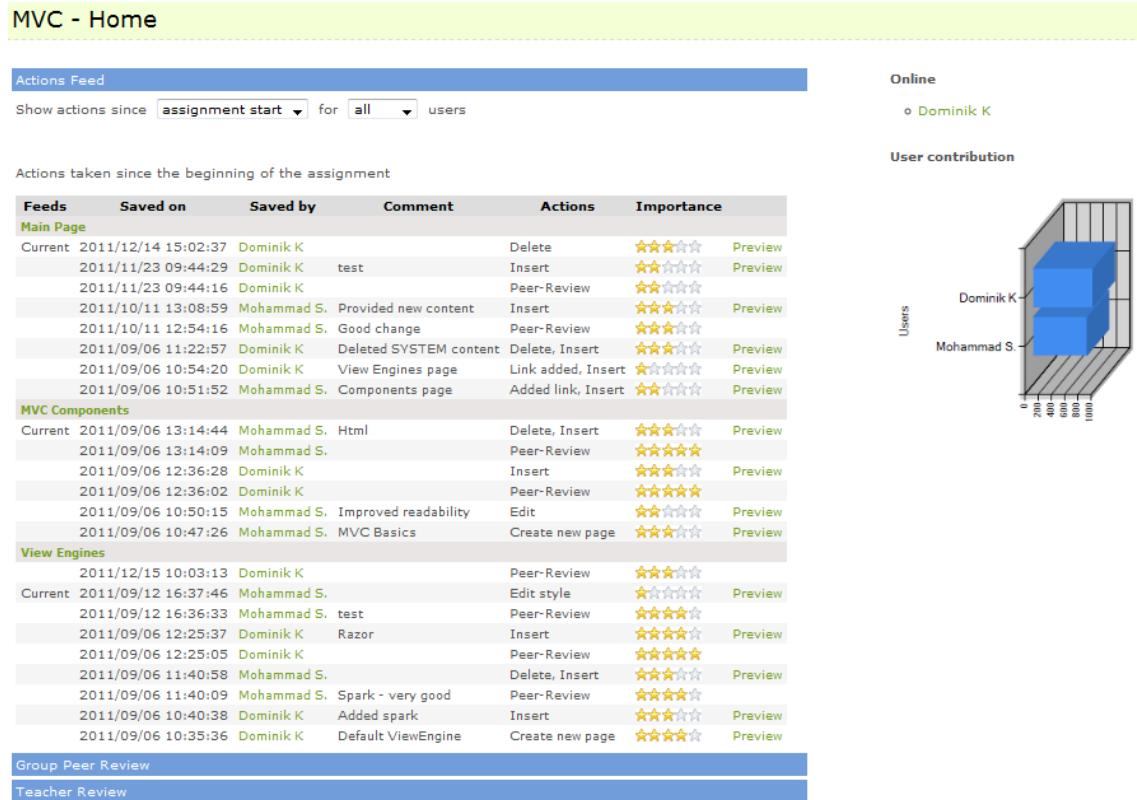


Figure 35: Assignment home page

The aim of the assignment homepage is to provide students with useful information in order to increase their task and social awareness. This is done by showing a so called “Actions Feed” table that lists the revision histories of all pages of the current assignment. So the students exactly know what their colleagues are doing. This “Actions feed” list also visualizes the results from the group intern assessment activities and offers links to preview the revision changes on the difference page (see subsection 8.2.2). Furthermore, this page shows social aspects on the right. This includes a list of online users to prevent communication and coordination problems and a chart that visualizes the individual contributions as reported in subsection 5.2.1.

Another page that visualizes the progress of student groups is the “Progress Charts” page that can be seen in Figure 36. This page visualizes the three phases of a collaborative writing assignment: the contribution phase, the assessment phase and the enhancement

## Chapter 8: Usage Point of View and Experimentations

phase. This means that the first chart shows the individual contributions until the first received group assessment. These group assessments are visualized in the second chart and can also be opened in a popup if the corresponding chart line is clicked. The third chart shows how much letters the students have been added after the group assessment phase and so gives an indication if the students used the group feedback for enhancing their work.

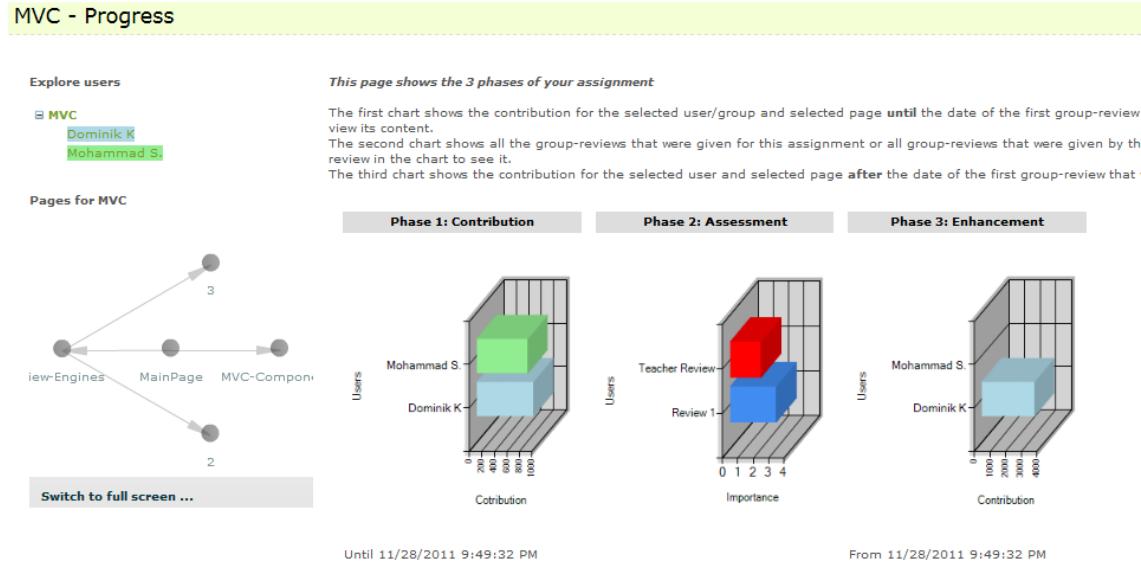


Figure 36: Progress charts page

### 8.2.4 Teacher Views

The first teacher view in Figure 37 shows the inline teacher feedback on the edit page. Here the teacher can use the WYSIWYG editor to tag parts of the Wiki pages in order to provide feedback (check spelling, check style, extend text, shorten text, shorten text, rewrite text, check reference, check link).

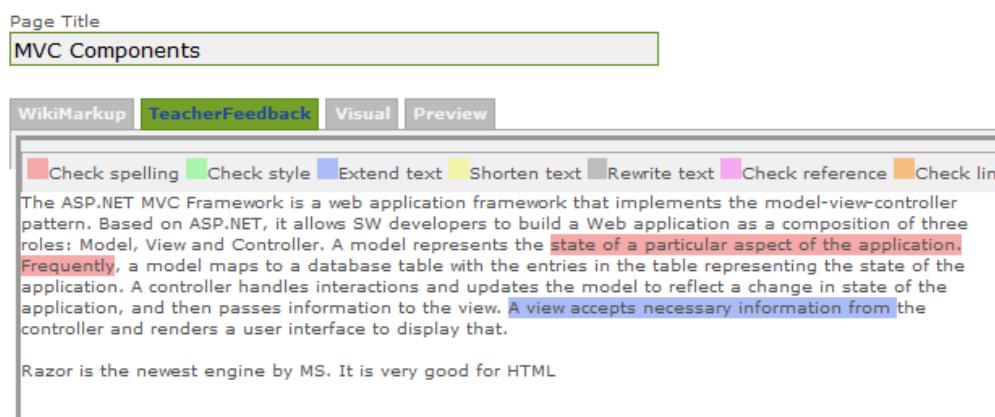


Figure 37: Inline teacher feedback on the edit page

## Chapter 8: Usage Point of View and Experiments

The meanings of the taggings can be seen in the colour-legend above (e.g. a blue tagging means that the coloured text should be extended). The main page for the teacher is visualized in Figure 38.

# Teachers view for: AspNet

Explore groups

- MVC
  - Dominik K.
  - Mohammad S.
- WebForms
  - Daniel B.
  - Johanna P.

Pages for MVC

```
graph TD; ViewEngines[View-Engines] -- 3 --> MainPage[MainPage]; ViewEngines -- 2 --> MVCComponents[MVC-Components];
```

[Switch to full screen ...](#)

Page Navigation [Back](#) [Main](#)

Revision player for group MVC on page View-Engines

[Play](#) Revision 6 [<"><<](#)

[<"><](#)

[<">>](#)

[<">>>](#)

Loop  Colored contribution

The mainstream view engine used in the ASP.NET MVC Framework is the Web Forms view is not very HTML friendly. By default, the view engine in the MVC framework uses regular interface pages onto which the data is composed. However, different view engines can be ASP.NET postback model, any interactions are routed to the controllers using the ASP.NET REST-friendly URLs.

Also possible is Spark which is a view engine for the ASP.NET MVC (and the Castle Project

Razor is the newest Engine by MS

Mapping Java classes to database tables is accomplished through the configuration of an using an XML file, Hibernate can generate skeletal source code for the persistence classes. Hibernate can use the XML file or the annotations to maintain the database schema.

Facilities to arrange one-to-many and many-to-many relationships between classes are p between objects, Hibernate can also manage reflexive associations where an object has a its own type.

Hibernate supports the mapping of custom value types. This makes the following scenario

Overriding the default SQL type that Hibernate chooses when mapping a column to a prop Mapping Java Enum to columns as if they were regular properties.

Useful information for group MVC on page View-Engines

Attribute	Overall	Page
Contribution [% Letters]	100	100
Contribution [# Letters]	5093	4032
No. Revisions [%]	100	100
No. Revisions [#]	17	6
No. Inserts [#]	12	4
No. Deletes [#]	4	1
No. Edits [#]	3	1
No. Style-edits [#]	2	1

Chart Type: [Column Chart](#)

Ratio [%]

Ratio [Content]

2011/09/06 10:35:36  
2011/09/06 10:40:38  
2011/09/06 11:40:58  
2011/09/06 12:25:37  
2011/09/12 16:37:46  
2012/01/19 11:44:57

Date	Mohammad S. (%)	Dominik K. (%)
2011/09/06 10:35:36	100	100
2011/09/06 10:40:38	100	100
2011/09/06 11:40:58	100	100
2011/09/06 12:25:37	100	100
2011/09/12 16:37:46	100	100
2012/01/19 11:44:57	100	100

Figure 38: Individual contribution visualization on the teacher's view

This page is similar to the contribution visualization page of the student (see subsection 8.2.3), but shows more information. One difference is that the teacher also sees the students of all the groups of the current assignment. Moreover, the individual contributions of each student are visualized with a specific colour in the revision player in the centre of the page to let the teacher know which student has provided which parts of the pages. This revision player also lets the teacher navigate through the complete page histories to better understand how the pages have been created. To increase these understandings, the bottom of the page shows further quantitative information to the teacher.

## Chapter 8: Usage Point of View and Experimentations

Another teacher view is used for entering final grades and is illustrated in Figure 39. This page uses the importance scorings from the group assessment phase to suggest a final grade to the teacher. The teacher can use this final score or can change it based on the individual contributions. In integrated settings this page also shows the associated concepts/taxons that can be saved back to the LMS (see subsection 7.3.3).

Final Grades for: AspNet

Explore groups	Gradings [%]	
<input type="checkbox"/> MVC Dominik K. Mohammad S.	Avg Peer-Score: 55	
<input type="checkbox"/> WebForms Daniel B. Johanna P.	Avg Teacher-Score: 54	
	Peer-Score Weighting <input type="text" value="50"/>	
	Full Score: <input type="text" value="54"/>	
	<input type="button" value="Save Score"/>	
Taxon ID	Taxon Name	Taxon Value
0	AspNet	52
0	JavaScript	40
0	JQuery	65

Figure 39: Final grades page

The next section describes the experimentations that have been administrated as part of this project.

## 8.3 Experimentations

As reported in the non-function requirements in subsection 6.2.2, it is necessary to test the Wiki functionalities by experimentations to find bugs and possible improvements. Furthermore, these experimentations should check if the implemented Wiki pages really support the students and teachers in processing a collaborative writing assignment. This should answer the following question: “Do the Wiki functionalities really increase the motivational aspects and the task awareness of students?” Thus, the aim of this section is to present an overview of the experimentations that have been administrated as part of this project and to explain their results and findings.

### 8.3.1 Experimentations Overview

This subsection gives an overview of the experimentations that have been conducted as part of this project. The details of these five experiments are listed in Table 8 in chronological order.

Location	Course	Scheduling	# Students	Specifics
TU Graz	Human Computer Interaction (HCI)	May 2011	21	Scientific paper writing
TU Graz	Scientific Working Seminar	July 2011	8	Lifelong learning seminar
Curtin University	Information Technology Seminar	August 2011	15	Controlled classroom environment
TU Graz	Information Search and Retrieval (ISR)	October – December 2011	26	Included an enhancement phase
KFU Graz	Biological Psychology	March – April 2012	31	IWT Integration

Table 8: Experimentations overview

The experimentations in Table 8 were all administrated using the same procedure. After the students were assigned to a topic and a group, they were asked to answer a pre-questionnaire to find out about their previous experience with Wikis, self and peer assessment activities and collaborative writing in general. In the second phase the students started to communicate and collaborate using the Wiki to create their essays. Thus, they used the assignment home page to check the actions from their colleagues in order to know the current state of the essay. Moreover, they used internal self and peer assessment to inform their colleagues about their own intentions and to give helpful feedback. When the students had finished their collaborative essays, they were asked to start the group assessment phase. Here the students had to group assess the essays from other groups using an assessment rubric (see subsection 8.2.2). After this phase, the students had the possibility to enhance their work based on the feedback of the other groups before the tutors did the final assessment using the same rubric. In the last phase the students and tutors were asked to fill out a post-questionnaire to give feedback about the usability and stability of the Wiki and to give information about their motivational aspects when using the different Wiki functionalities. (AL-Smadi, Hoefler, & Guetl, 2011a, 2011b; AL-Smadi, Hoefler, Wesiak, & Guetl, 2012)

During the experimentations the author of this thesis was responsible to set up the necessary environments in the Wiki and for technical support in case of questions and problems. Furthermore he had the role of a tutor in the experimentation for the “Information Search and Retrieval” course and was responsible for giving tutor group assess-

ments. The rest of the research group were responsible for the administrative support and for creating and analysing the questionnaires. The next subsection reports the results and findings of the experimentations and describes how these results have influenced the development process of the Wiki.

### 8.3.2 Results and Findings

This subsection illustrates the results and findings of the experimentations according to AL-Smadi, Hoefler and Guel (2011a), AL-Smadi, Hoefler and Guel (2011b) and AL-Smadi, Hoefler Wesiak and Guel (2012). These results are mainly based on the student questionnaires of the first administrated experimentation done for the TU Graz course “Human Computer Interaction”, because other results have not been published until now. However, the remainder of this subsection not only shows the positive and negative results of this experimentation, but also outlines general issues identified by the teachers of the other experimentations.

#### Positive student findings

- Internal self assessment supports students to efficiently reflect their own work and to give information to their group members about their actions.
- Internal peer assessment helps students to track the actions of their colleagues in order to increase task and social awareness.
- External group assessment supports students in learning from the work of other groups to better enhance their own work. Furthermore, the students think that assessment rubrics are a great tool for reviewing writing assignments.
- The assignment home page helped students to coordinate their tasks with their group members to increase task and social awareness. Moreover, the progress of their peers acts as great motivational aspects.
- In general students like the idea of the Wiki and especially the assignment home page. They also think that the Wiki functionalities can be learned very fast by most people.

#### Negative student findings

- Internal self assessment is thought by some students to be stressful in case of minor changes, but this functionality only makes sense if it is used for every change.
- Internal peer assessment needs to be optional because not every action of the peers needs to be reviewed. To realize this, four different peer assessment modes have

been implemented (mandatory, optional, based on indentations and based on revision numbers).

- External group assessment was only possible on the level of namespaces and not on the level of pages. Furthermore, students thought that the assessment rubric has too many fields and that not every field should be mandatory.
- The assignment home page needs a lot of time to load because the contribution chart needs a lot of data. To solve this, the chart data is now loaded asynchronously to increase the performance.
- In general, students thought that the Wiki functionalities are not self-explaining when used for the first time and so more explanations were added. Some students also complained because the edit page needs a lot of time for saving and so the slow automatic database index calculation of MySQL was turned off. This can now be done by the administrator from time to time. Moreover, the students asked for a way to change their password which also has been implemented.

### General teacher issues

- The assignment home page should also show the authors of the external group assessments.
- Teachers need a page that visualizes the three phases of the collaborative writing assignment (contribution, assessment and enhancement phase as seen in subsection 8.2.3).
- Teachers need a way to tag parts of Wiki pages with specific meanings in order to give more detailed feedback (see subsection 8.2.4).
- The assessment rubric of the group assessment has to support weightings because some elements are more important than others.
- The revision player on the teacher's view page should also show the associated internal peer assessments.
- The most important usage patterns such as login times, assessment processes and edit actions have to be saved in the database to know how the students are working.
- Teachers need a page for entering summative assessments based on the results from the group assessments.
- The page structure of a namespace should be visualized in order to better understand how the Wiki pages are linked together.

These identified issues and findings are also described in the sections 9.2 and 10.2.

## 8.4 Summary

The last chapter has given an overview of the development of the prototype and especially has shown the functionalities of the Wiki from the developer's point of view. The aim of this chapter is to illustrate these functionalities from the user's point of view. Furthermore, the experimentations that have been administrated as part of this project are explained. This includes also the experimentation findings and how they have influenced the development process of the Wiki.

The Wiki can be installed using its Visual Studio 2008 solution that can be found on the CD-ROM of this thesis (see Appendix: CD-ROM). This solution has to build and then can be used as a new Web site in the Microsoft Internet Information Services (IIS) manager. The CD-ROM also includes a SQL script to set up the MySQL database with some test data. This chapter also shows the Wiki functionalities from the user's point of view which can also be used as a user handbook. Thus, section 8.2 is separated into four functionality parts, the assignment authoring, the assessment and feedback activities, the student views and the teacher views, that are all visualized with screenshots and explanations of the tool.

As reported in the non-functional requirements in subsection 6.2.2, it is necessary to test the Wiki functionalities with experimentations to find bugs and possible improvements. This should answer the following question: Do the Wiki functionalities really increase the motivational aspects and the task awareness of students? The five experimentations that were administrated as part of this thesis all had the same procedure: first the students were asked to fill out a pre-questionnaire to know about their current knowledge state, then they were asked to contribute in the Wiki, to do the assessment tasks and to fill out a post-questionnaire in the last step. (AL-Smadi, Hoefler, & Guetl, 2011a, 2011b; AL-Smadi, Hoefler, Wesiak, & Guetl, 2012)

Based on the results from these post-questionnaires, some findings and issues regarding the Wiki functionalities have been identified. The students generally liked the idea of the Wiki and its assessment forms. Especially the internal self and peer assessment have increased the task and social awareness of the students and supported them in coordinating with their colleagues. Furthermore, they liked the idea of the external group assessment because they could use the knowledge gained during these activities for enhancing their own work. On the other hand, the students complained about the bad performance when loading the assignment home page or when trying to edit a page, but these problems have been solved. Moreover, the students and teachers mentioned that some functionality like a configurable internal peer assessment mode are still missing which helped the author of this

## Chapter 8: Usage Point of View and Experimentations

thesis a lot in extending the requirements catalogue. The next chapter illustrates the author's personal, technical user-oriented lessons learned.

## 9 Lessons Learned

The aim of this chapter is to state the author's personal, technical and user-oriented lessons learned during this master thesis. This should reflect the experiences, developments, problems and risks of this project in order to benefit from them in future projects.

### 9.1 Personal Point of View

The author of this thesis has benefited a lot from doing this project as a project fellow at the Institute for Information Systems and Computer Media (IICM) at the University of Technology, Graz. This has given many insights in the working process of a research group which has highly increased the author's scientific experiences. Beyond this, it was a great benefit to work together with colleagues from the psychological department to increase the own knowledge and to look at the things not always only from the technical point of view. Moreover, it has been a great experience to process this thesis as part of a project that is co-funded by the European Commission. This has given the author the possibility to get in touch with colleagues from Italy and to profit from their experience while working together on integration tasks.

Another important activity during this project has been the administration of the experimentations in order to test the Wiki in different learning settings. This has been a great experience for the author, especially when getting in touch with students to get to know the tutor or teacher view of such an experimentation or exercise. On the other hand, these experimentations have been very stressful because it was necessary to have all Wiki functionalities ready and tested until the start dates. Furthermore, some experimentations have needed a lot of student support, but have also led to a lot of interesting results and findings regarding the Wiki. The next section will describe the lessons learned from the technical point of view.

### 9.2 Technical Point of View

As reported in the previous chapters, ScrewTurn has been used as the basic Wiki software for this project. It has been a good experience for the author to become acquainted with the architecture and the code of a quite big open-source .NET project and to benefit from the software design that is based on the knowhow of an active user community. Especially the

## Chapter 9: Lessons Learned

implementation style of ScrewTurn's flexible plugin system has been very instructional. Furthermore, working with a Wiki system and designing enhancements for it highly increased the knowledge of the author in both ways, theoretical and technical. After the time-consuming literature research that has been achieved to understand the theoretically concepts and principles, it has been highly satisfying to see this theory get evolved in an own system.

On the contrary, the advantages and disadvantages of the server-side Asp.NET Web forms framework have been shown. This framework enables a rapid and event-driven development of Web applications that disguise the problems of this technology like its stateless architecture. However, this approach works very well for simple Web forms but for big pages the Postback and ViewState concepts of Asp.NET produce a lot of request and response data which can lead to a bad performance as seen in some experimentation. A better solution to handle more complex pages could be to use more client-side libraries like JQuery and to just call Web methods on the server via AJAX. The communication between client and server could be handled using a markup language like XML or JSON. As part of this project, this approach has been implemented using AJAX control libraries such as the ones by Microsoft or Anthem. These libraries seem to only update parts of the pages at Postbacks but although go through the whole page and control lifecycle which causes nearly the same server workloads. The next section will describe the lessons learned from the usage point of view.

### 9.3 Usage Point of View

The author of this thesis has benefited a lot from the user feedback received by the administrated experimentations. It was very satisfying to see that the implemented assessment functionalities really supported the users in writing their essays and in coordinating with their group members. Moreover, the students and teachers liked the assignment home page which was one of the hardest parts to implement. However, the author has also profited by the negative feedback. Here it was a great experience to see how small test data differs from real experimentation data in relation to the performance of the Wiki. These findings significantly increased the quality of the source code and the tool itself and will definitely influence the author's programming style in the future. The next chapter will sum up the most important concepts and findings of this thesis and will also describe approaches for future work.

## 10 Conclusion

The aim of this master thesis has focused on identifying, designing and implementing a solution approach for combining Computer-Supported, Collaborative Learning with E-Assessment. Thus, a Wiki system has been enhanced with assessment methods and other tools to support teachers and students in collaborative writing activities. The aim of this chapter is to sum up the most important concepts that have been identified in the literature research and to reflect the development process of the implemented Wiki prototype. Furthermore, possible approaches for future work are illustrated.

### 10.1 Summary

Computer-Supported, Collaborative Learning (CSCL) is a research area that describes approaches how collaborative learning can be supported with modern computer and information technologies. These computer-supported environments (e.g. a Wiki system) should help the learners in bringing together their individual points of view to create new collaborative content and knowledge (Ma, 2008). In order to efficiently realize this approach in modern learning environments, it is important to combine it with E-Assessment methods that are used to define, evaluate, rate, document and feedback learning processes before, during and after the teaching period with the help of modern technologies (Brahm & Seufert, 2007). The literature research has shown that such combinations are facing some problems: the lack of awareness, coordination problems, communication problems and motivational aspects, which includes also the lack of assessment activities.

Based on these identified issues, an existing Wiki system has been designed, implemented and tested within different experimentations. Beside these typical functional and non-functional tool requirements, it has been a main goal of this project that the Wiki should be usable in two ways, as a standalone Web application and as an integrated Web tool within existing Learning Management Systems (LMS) such as IWT or SOFIA. On the basis of these requirements, ScrewTurn has been chosen to be enhanced and extended as part of this project. ScrewTurn is based on the Microsoft Asp.NET Web framework and uses a MySQL database as data backend. Multiple experimentations have been administrated in order to test the Wiki and to get feedback from the users. The results and findings have shown that the users liked the idea of the Wiki, especially the assessment forms and the assignment home page, but complained about missing features and some performance

## Chapter 10: Conclusion

issues that have already been solved in most cases. The next section illustrated some possible approaches for future work.

## 10.2 Future Work

There are two possible ways in enhancing the system that have been identified during the development and experimentation processes: increasing the performance of the Wiki and extending the functionalities. In case of performance issues, the server-side Asp.NET controls could be exchanged with more lightweight client-side controls as offered by JQuery that use lightweight AJAX calls to access the server. Furthermore, some data could be cached in the database in order to prevent complicate calculations. This problem especially occurs in integrated environment when lots of Web services of LMSs are used.

Regarding new functionalities, the static rubric that is used for assessing the collaboratively created essays could be exchanged with a dynamic one that is saved in the database and that can be authored by teachers. Moreover, these dynamically created rubric fields could also be linked to specific concepts in order to enhance the summative assessment process (see subsection 7.3.3). Another issue which has been mentioned by some teachers is that the Wiki does not offer an overview page for the group assessment activities. This makes it hard for the teacher to quickly check if all students have finished their assigned group assessments. In addition to this, a PDF print of complete Wiki namespaces would support the users in their assessment tasks. This would have a big advantage if there are a lot of interlinked pages that have to be followed to read through the whole writing assignment.

## 11 Bibliography

- ALICE. (2009). *Adaptive Learning via Intuitive/Interactive Collaborative and Emotional Systems*. Retrieved 04 10, 2012, from <http://aliceproject.eu/>
- AL-Smadi M., Guetl, C., Caballe, S., & Dunwell, I. (2012). *D5.2.2. Enriched Learning Experience v2*. Retrieved from  
<alice.crmpa.unisa.it/members/home/delintereport/Deliverable%20%20Internal%20Report/Enriched%20Learning%20Experience%20v2.pdf>
- AL-Smadi, M., & Guetl, C. (2008). Past, Present and Future of e-Assessment-Towards a Flexible e-Assessment System. *Proceeding of ICL2008, Villach, Austria.*
- AL-Smadi, M., & Guetl, C. (2011). Service-oriented flexible and interoperable assessment: towards a standardised e-assessment system. *International Journal of Continuing Engineering Education and Life Long Learning*, 21(4), 289-307.
- AL-Smadi, M., Guetl, C., & Chang, V. (2011). Addressing e-Assessment practices in e-Learning Activities: A Review. In S.-M. Barton, J. Hedberg, & K. Suzuki (Ed.). (pp. 448-459). Melbourne, Australia: AACE.
- AL-Smadi, M., Hoefler, M., & Guetl, C. (2011a). Enhancing Wikis with Visualization Tools to Support Groups Production Function and to Maintain Task and Social Awareness. *International Conference on Interactive Computer Aided Blended Learning (ICBL 2011)*, (p. 6). Antigua Guatemala.
- AL-Smadi, M., Hoefler, M., & Guetl, C. (2011b). Integrated and enhanced e-assessment forms for learning: Scenarios from ALICE project. *IEEE*, (pp. 626-631).
- AL-Smadi M., Hoefler M., Wesiak G., & Guetl, C. (2012). Complex Learning Resources Integrated with Emerging Forms of e-Assessment: an Empirical Study. *IEEE International Conference on Advanced Learning Technologies (ICALT 2012)*, Rome 7-10 July 2012, ACCEPTED
- Andrade, H. (2000). Using rubrics to promote thinking and learning. *Educational Leadership*, 57(5), 13-19.
- Aranda, N. (2005). *A Brief History of E-learning and Distance Education*. Retrieved from Ezine Articles: <http://www.ezinearticles.com/?A-Brief-History-of-E-learning-and-Distance-Education>

## Chapter 11: Bibliography

- Assembla. (2012). *Assembla workspaces to accelerate software teams*. Retrieved 04 10, 2012, from <http://www.assembla.com/>
- Berger, D. (2011). *Supporting Tool for Moderation in the Grading Process of Summative Assessments*. Master's thesis, University of Technology Graz.
- Biuk-Aghai, R., Kelen, C., & Venkatesan, H. (2008). Visualization of interactions in collaborative writing. *IEEE*, (pp. 97-102).
- Bostock, S. (2000). Student peer assessment. *Keele University, Learning Technology Documents, December*.
- Brahm, T., & Seufert, S. (2007). "Ne(x)t Generation Learning": E-Assessment und E-Portfolio: halten sie, was sie versprechen? Tech. rep.
- Brown, J., & Adler, R. (2008). Open education, the long tail, and learning 2.0. *Educause review*, 43(1), 16-20.
- Buzzetto-More, N. A., & Alade, J. A. (2006). Best Practices in e-Assessment. *JITE*, 5, 251-269.
- Challborn, C., & Reimann, T. (2005). Wiki Products: a comparison. *The International Review of Research in Open and Distance Learning*, 6(2).
- Clariana, R., & Wallace, P. (2002). Paper--based versus computer--based assessment: key factors associated with the test mode effect. *British Journal of Educational Technology*, 33(5), 593-602.
- Crisp, G. (2007). *The e-Assessment Handbook*. London: Continuum.
- Crisp, G. (2009). Towards Authentic e-Assessment Tasks. In G. Siemens, & C. Fulford (Ed.). (pp. 1585-1590). Honolulu, HI, USA: AACE.
- CSharp-Source. (2010). *Open Source Wiki Engines in C-Sharp*. Retrieved 04 10, 2012, from <http://csharp-source.net/open-source/wiki-engines>
- Dillenbourg, P., & Fischer, F. (2007). Basics of Computer-Supported Collaborative Learning. *Zeitschrift fuer Berufs- und Wirtschaftspaedagogik*, 21, 111-130.
- Dochy, F., Segers, M., & Sluijsmans, D. (1999). The use of self-, peer and co-assessment in higher education: a review. *Studies in Higher education*, 24(3), 331-350.
- Dror, I. (2008). Technology enhanced learning: The good, the bad, and the ugly. *Pragmatics & Cognition* #38, 16(2), 215-223.

## Chapter 11: Bibliography

- Duffy, P., & Bruns, A. (2006). The use of blogs, wikis and RSS in education: A conversation of possibilities.
- Earl, L. (2003). *Assessment as learning: Using classroom assessment to maximize student learning*. Corwin Pr.
- Ebner, M. (2009). Technology Enhanced Learning. *Technology Enhanced Learning*.
- Ebner, M., & Schoen, S. (2011). L3T: Lehrbuch fuer Lernen und Lehren mit Technologien (ger). *Identity, 2011*, 7-26.
- E-Teaching. (2010). *CBT/Computer-Based Training*. Retrieved 04 10, 2012, from <http://www.e-teaching.org/glossar/cbt>
- Forte, A., & Bruckman, A. (2007). Constructing text: : Wiki as a toolkit for (collaborative?) learning. In A. Désilets, & R. Biddle (Ed.). (pp. 31-42). ACM.
- Gonzalez, J., Rodriguez, M., Nistal, M., & Rifon, L. (2009). Reverse OAuth: A solution to achieve delegated authorizations in single sign-on e-learning systems. *Computers & Security, 28*(8), 843-856.
- Guetl, C. (2008). Moving towards a Fully Automatic Knowledge Assessment Tool. *iJET, 3*(1).
- Guetl, C., Lankmayr, K., Weinhofer, J., & Hofler, M. (2010). Enhanced approach of automatic creation of test items to foster modern learning setting., (pp. 4-5).
- Harlen, W. (2005). Teachers' summative practices and assessment for learning--tensions and synergies. *Curriculum Journal, 16*(2), 207-223.
- Hayes, T., & Ge, X. (2008). The effects of computer-supported collaborative learning on students' writing performance. (pp. 335-341). International Society of the Learning Sciences.
- Helic, D., Maurer, H., & Scerbakov, N. (2000). Web based training: What do we expect from the system., (pp. 1689-1694).
- Higgins, C. A., & Bligh, B. (2006). Formative computer based assessment in diagram based domains. In R. Davoli, M. Goldweber, & P. Salomoni (Ed.). (pp. 98-102). ACM.
- Jones, P., Kolloff, M., & Kolloff, F. (2010). Best Practices to Support and Assess Learning Using Wikis.

## Chapter 11: Bibliography

- Khandaker, N., & Soh, L. (2010). ClassroomWiki: A collaborative wiki for instructional use with multiagent group formation. *Learning Technologies, IEEE Transactions on*, 3(3), 190-202.
- Kimmerle, J., Moskaliuk, J., & Cress, U. (2009). Understanding learning: the Wiki way. In D. Riehle, & A. Bruckman (Ed.). ACM.
- Kraut, R. E., Galegher, J., Fish, R. S., & Chalfonte, B. L. (1992). Task requirements and media choice in collaborative writing. In *Human-Computer Interaction* (Vol. 7, pp. 375-407).
- Lai, Y., & Ng, E. (2011). Using wikis to develop student teachers' learning, teaching, and assessment capabilities. *The Internet and Higher Education*, 14(1), 15-26.
- Leuf, B., & Cunningham, W. (2001). *The Wiki Way: Quick Collaboration on the Web* (Pap/Cdr ed.). Addison-Wesley Longman, Amsterdam.
- Ma, W. W. (2008). Computer Supported Collaborative Learning and Social Creativity of Fashion Design. *Journal of Information, Information Technology and Organizations*, 3.
- Macdonald, J. (2003). Assessing online collaborative learning: process and product. *Computers & Education*, 40(4), 377-391.
- Microsoft. (2012). *Microsoft Asp.NET. Web-Development with Power, Productivity and Speed*. Retrieved 04 10, 2012, from <http://www.asp.net/>
- MSDN. (2012). *Microsoft Developer Network Online Library*. Retrieved 04 10, 2012, from <http://msdn.microsoft.com/en-us/>
- Myers, E. (1986). An O (ND) difference algorithm and its variations. *Algorithmica*, 1(1), 251-266.
- MySQL. (2012). *MySQL. The world's most popular open source database*. Retrieved 04 10, 2012, from <http://mysql.com/>
- NDoc. (2005). *NDoc Code Documentation Generator for .NET*. Retrieved 04 10, 2012, from <http://ndoc.sourceforge.net/>
- Nichols, M. (2008). E-Learning in context. *E-Primer series*, 1.
- Noel, S., & Robert, J.-M. (2004). Empirical Study on Collaborative Writing: What Do Co-authors Do, Use, and Like? *Computer Supported Cooperative Work (CSCW)*, 13(1), 63-89.

## Chapter 11: Bibliography

- Nohr, H. (2005). Konzeption und Evaluation von CSCL-basierten Seminaren. In J. M. Haake, U. Lucke, & D. Tavangarian (Ed.). 66, pp. 69-80. GI.
- NUnit. (2007). *.NET Unit Testing Framework*. Retrieved 04 10, 2012, from <http://www.nunit.org/>
- Peyton, J. K. (1990). Technological Innovation Meets Institution: Birth of Creativity or Murder of a Great Idea? *Computers and Composition*, pp. 15-32.
- Rama, J., & Bishop, J. (2006). A survey and comparison of CSCW groupware applications. (pp. 198-205). Republic of South Africa: South African Institute for Computer Scientists and Information Technologists.
- Roberts, T. (2006). *Self, peer, and group assessment in e-learning*. Information Science Publishing.
- Salmon, G. (2002). *E-tivities: The Key to Active Online Learning*. Taylor & Francis Ltd.
- ScrewTurn. (2012). *ScrewTurn Asp.NET 3.5 based Wiki System*. Retrieved 04 10, 2012, from <http://www.screwtturn.eu/>
- Specht, M. (2009). *Learning in a technology enhanced world*.
- Stahl, G., Koschmann, T., & Suthers, D. (2006). Computer-supported collaborative learning: An historical perspective. In R. K. Sawyer (Ed.). (pp. 409-426). Cambridge University Press.
- Strijbos, J. (2011). Assessment of (computer-supported) collaborative Learning. *Learning Technologies, IEEE Transactions on*(99), 1-1.
- Tammaro, S. G., Mosier, J. N., Goodwin, N. C., & Spitz, G. (1997). Collaborative Writing Is Hard to Support: A Field Study of Collaborative Writing. *Computer Supported Cooperative Work*, 6(1), 19-51.
- TLT. (2009). Retrieved 04 10, 2012, from The TLT Group: Teaching, Learning and Technology, Rubrics: <http://www.tltgroup.org/resources/rubrics.htm>
- w3schools. (2012). *w3schools.com. The world's largest web development site*. Retrieved 04 10, 2012, from <http://www.w3schools.com/>
- Wang, R., Chen, S., & Wang, X. (2012). Signing Me onto Your Accounts through Facebook and Google: a Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services. *Signing Me onto Your Accounts through Facebook and*

## Chapter 11: Bibliography

*Google: a Traffic-Guided Security Study of Commercially Deployed Single-Sign-On Web Services.*

Xiao, Y., & Lucking, R. (2008). The impact of two types of peer assessment on students' performance and satisfaction within a Wiki environment. *The Internet and Higher Education*, 11(3-4), 186-193.

## 12 Appendix: CD-ROM

- Source code of the Wiki
- MySQL file for setting up the database
- PDF version of this document
- Documents of Chapter 11: Bibliography as PDF files
- Screenshots of the Wiki
- Video that shows the functionalities of the Wiki from the user's point of view
- Marketing video which was created as part of the ALICE project