



UNIVERSITÄT DUISBURG-ESSEN
FAKULTÄT FÜR INGENIEURWISSENSCHAFTEN
ABTEILUNG INFORMATIK UND ANGEWANDTE KOGNITIONSWISSENSCHAFT

Bachelorarbeit

Entwicklung eines Moodle Plug-Ins für kollaboratives Concept Mapping

Henning Traulsen
Matrikelnummer: 2243000

UNIVERSITÄT
D U I S B U R G
E S S E N

Abteilung Informatik und angewandte Kognitionswissenschaft
Fakultät für Ingenieurwissenschaften
Universität Duisburg-Essen

22. September 2014

Betreuer:

Prof. Dr. H. U. Hoppe
Prof. Dr.-Ing T. Zesch
Tobias Hecking, M.Sc.

Inhaltsverzeichnis

Inhaltsverzeichnis.....	iii
Abbildungsverzeichnis.....	vii
1. Einleitung	1
1.1. Motivation	1
1.2. Aufgabenstellung.....	1
1.3. Aufbau der Arbeit	2
2. Grundlagen.....	3
2.1. Concept Maps.....	3
2.1.1. Einführung zu Concept Maps.....	3
2.1.2. Concept Maps und Mind Maps.....	4
2.1.3. Kollaborative Concept Maps.....	5
2.2. Kooperatives Lernen	6
2.3. Computer Supported Collaborative Learning (CSCL)	6
2.3.1. Einführung	6
2.3.2. Groupware.....	7
2.3.3. Mehrbenutzereditoren.....	7
2.3.4. Sitzung	8
2.3.5. Geteilter Kontext.....	8
2.3.6. Herausforderungen	8
2.4. Architekturen.....	9
2.4.1. Coupling.....	9
2.4.2. Zentrale Architektur	10
2.4.3. Replizierte Architektur.....	10
2.4.4. Verteilte Architektur.....	10
2.4.5. Hybride Architektur.....	11

2.5. Moodle	11
2.6. TogetherJS	11
2.7. Go-Lab	12
3. Ansatz.....	13
3.1. Einführung	13
3.2. Anforderungen an die Synchronisierung.....	13
3.2.1. Synchronisierung von Benutzeraktionen.....	13
3.2.2. Synchronisierung neuer Teilnehmer	14
3.2.3. Wahrung der Konsistenz.....	14
3.2.4. Zusammenfassung.....	14
3.3. Anforderungen an das Plug-In	15
3.3.1. Einführung	15
3.3.2. Sitzungsmanagement	15
3.3.3. Asynchrone Bearbeitung	15
3.3.4. Unterstützung von Moodle Gruppen	15
3.3.5. Einstellungsmöglichkeiten	16
3.3.6. Protokollierung	16
3.3.7. Kommentarfunktion	16
3.3.8. Darstellung.....	16
3.4. Zusammenfassung aller Anforderungen	18
3.5. Architektur des Plug-Ins	18
3.5.1. Überblick.....	19
3.5.2. Synchronisierung mit TogetherJS.....	20
4. Implementierung	23
4.1. Speicherung von Concept Maps in der Moodle Datenbank.....	24
4.2. Protokollierung sowie Ausgabe von Benutzeraktionen	26
4.3. Verschiedene Sichten für die Nutzer des Plug-Ins	26
4.4. Initialisierung der TogetherJS Sitzung	27
4.5. Synchronisierung von Benutzeraktionen	28
4.6. Behandlung von neuen Teilnehmern in einer Sitzung	28
4.7. Umgang mit Übertragungsfehlern und konkurrierenden Aktionen	29

5. Evaluation.....	31
5.1. Anwendungsfälle	31
5.1.1. Lehrende - Innerhalb des Bearbeitungszeitraums	32
5.1.2. Studenten - Innerhalb des Bearbeitungszeitraums.....	34
5.1.3. Lehrende - Außerhalb des Bearbeitungszeitraums.....	36
5.1.4. Studenten - Außerhalb des Bearbeitungszeitraums	37
5.2. Mögliche Probleme mit TogetherJS.....	37
6. Zusammenfassung und Ausblick	39
6.1. Zusammenfassung	39
6.2. Ausblick	39
Literaturverzeichnis.....	41
A. Eidesstattliche Erklärung zur Bachelorarbeit	43

Abbildungsverzeichnis

<i>Abbildung 1: Erstellung von Concept Maps</i>	<i>4</i>
<i>Abbildung 2 Concept Map und Mind Map.....</i>	<i>5</i>
<i>Abbildung 3 Time/Space Matrix.....</i>	<i>7</i>
<i>Abbildung 4 Architektur des Plug-Ins</i>	<i>19</i>
<i>Abbildung 5 Synchronisierung über TogetherJS</i>	<i>20</i>
<i>Abbildung 6 Nachrichtenaustausch über TogetherJS</i>	<i>21</i>
<i>Abbildung 7 Detaillierter Aufbau des Plug-Ins</i>	<i>23</i>
<i>Abbildung 8 Überblick über die Metadaten</i>	<i>25</i>
<i>Abbildung 9 Speichern einer Concept Map</i>	<i>25</i>
<i>Abbildung 10 Schematische Darstellung der Tabellen des Plug-Ins</i>	<i>26</i>
<i>Abbildung 11 Nachrichtenaufbau und Aufbau des activityStreamObjects.....</i>	<i>27</i>
<i>Abbildung 12 Konfiguration des Plug-Ins.....</i>	<i>32</i>
<i>Abbildung 13 Auswahl der Gruppe.....</i>	<i>32</i>
<i>Abbildung 14 Lehrende - Innerhalb des Bearbeitungszeitraums</i>	<i>33</i>
<i>Abbildung 15 Ein Student bearbeitet die Concept Map.....</i>	<i>34</i>
<i>Abbildung 16 Zweiter Student tritt der Sitzung bei</i>	<i>34</i>
<i>Abbildung 17 Drei Studenten bearbeiten die Concept Map.....</i>	<i>35</i>
<i>Abbildung 18 Lehrend - Außerhalb des Bearbeitungszeitraums</i>	<i>36</i>
<i>Abbildung 19 Studenten - Außerhalb des Bearbeitungszeitraums.....</i>	<i>37</i>

1. Einleitung

1.1. Motivation

Concept Maps sind eine Methode zur strukturierten Visualisierung von bestehendem Wissen. Dabei werden einzelne Begriffe, die Konzepte, über Verbindungen miteinander vernetzt. Durch diese Wissensstrukturierung unterstützen Concept Maps das reflektierende Denken (Novak 1998, in Xie & Sharma 2011). So können Lernende komplexe Themen leichter erschließen, und es können Wissenslücken einfacher erkannt werden (Gurlitt & Nückles, 2010). Die kollaborative Bearbeitung von Concept Maps kann als Wissensbasis unter Studierenden dienen. Die gemeinsame Bearbeitung erleichtert auch die Aktivierung von bestehendem Wissen (Ostwald 1996, in Gao et al. 2007), durch Kommunikation können Ideen ausgetauscht und von anderen Teilnehmern aufgegriffen und erweitert werden.

Moodle ist eine modulare Lehrplattform und kann durch sogenannte Module um neue Funktionen erweitert werden. Solche Plug-Ins sind unter anderem Aktivitäten an denen Studierende innerhalb eines Kurses teilnehmen können. Es gibt bereits Anwendungen zur kollaborativen Erstellung von Concept Maps¹, sowie Moodle Plug-Ins², welche die Erstellung von Concept Maps ermöglichen. Jedoch gibt es noch kein Moodle Plug-In welches die kollaborative Bearbeitung von Concept Maps im Rahmen von Kursen in Moodle ermöglicht.

So ein Plug-In hätte den Vorteil, dass es direkt als Aktivität in Moodle eingesetzt werden könnte. Auch können so Funktionen von Moodle, wie zum Beispiel die Zuordnung von Studenten zu Gruppen ausgenutzt werden. In diesem Fall bearbeiten Studenten, pro Aktivität und Gruppe, immer eine Instanz einer Concept Map. Diese können gespeichert und zu späteren Zeitpunkten weiter bearbeitet werden. So wäre im Kontext von Moodle die synchrone sowie asynchrone Bearbeitung von Concept Maps möglich.

1.2. Aufgabenstellung

In dieser Arbeit soll ein Moodle Plug-In erstellt werden, welches die kollaborative Bearbeitung von Concept Maps ermöglicht. Im Kontext von Moodle soll eine gewisse Konfigurierbarkeit des Plug-Ins ermöglicht werden. Dazu gehört die Auswahl ob Moodle-Gruppen unterstützt werden sollen und das Festlegen eines Abgabedatums. Studierende sollen nun innerhalb ihrer Gruppen gemeinsam an derselben Concept Map arbeiten können. Diese können gespeichert und zu späteren Zeitpunkten bis hin zum Abgabedatum weiter bearbeitet werden. Weiterhin wird die Protokollierung von Benutzeraktionen unterstützt, welche Lehrende einsehen können. Auch wird Lehrenden die Möglichkeit geboten Feedback zu den abgegebenen Concept Maps zu hinterlassen.

¹ CmapTools <http://cmap.ihmc.us/>

² <http://blog.matbury.com/2013/08/15/free-and-open-source-concept-map-app-for-moodle/>

1. Einleitung

1.3. Aufbau der Arbeit

Kapitel 2 erläutert zunächst die theoretischen Grundlagen dieser Arbeit. Es werden dabei Begriffe und Konzepte vorgestellt, auf die im Laufe der Arbeit aufgebaut wird.

Im Anschluss sollen in Kapitel 3 zum Ansatz, aus der Aufgabenstellung und den Grundlagen alle Anforderungen an das zu entwickelnde Plug-In herausgearbeitet werden. Auch wird hier ein Überblick über die Architektur des Plug-Ins geboten.

Im 4. Kapitel wird die Implementierung der im Ansatz vorgestellten Funktionen besprochen. Kapitel 5 soll das entwickelte Plug-In vorstellen und kritisch in Hinblick auf die Funktionalität bewerten.

Im 6. Kapitel erfolgt eine Zusammenfassung der Arbeit und es wird ein Ausblick auf weitere Verbesserungen sowie die generelle mögliche Weiterentwicklung gegeben.

2. Grundlagen

Dieses Kapitel soll die theoretischen Hintergründe dieser Arbeit herausarbeiten und grundlegende Begriffe, auf die im Verlauf der Arbeit Bezug genommen werden definieren. Es wird zunächst das Konzept von Concept Maps eingeführt, auf Besonderheiten eingegangen und das kollaborative Bearbeiten von Concept Maps wird evaluiert. Anschließend wird das Feld der Computer-Supported-Collaborative-Work vorgestellt. Dabei werden wichtige Aspekte herausgearbeitet, sowie Architekturen zur Erstellung von kollaborativen Anwendungen erläutert.

Wie schon in der Einführung hervorgehoben, wird das hier entwickelte Moodle Plug-In Anwendung in der Lehre finden. Deshalb werden weitere Beschreibungen und Erläuterungen von Konzepten bzw. Begriffen auch in erster Linie in Bezug auf deren Anwendung in der Lehre nehmen.

2.1. Concept Maps

2.1.1. Einführung zu Concept Maps

Concept Maps visualisieren Begriffe und deren Zusammenhänge durch Konzepte und Verbindungen zwischen diesen Konzepten. Eine Concept Map lässt sich dabei als gerichteter Graph beschreiben. Konzepte werden hier durch Knoten des Graphs dargestellt und vertreten durch eine Beschriftung die Begriffe. Die Kanten des Graphs stellen die Verbindungen dar und geben die Leserichtung vor. Die Beschriftung der Kanten ermöglicht es die Zusammenhänge genauer darzustellen.

Die Erstellung einer Concept Map kann nach (Gurlitt & Nückles, 2010) in vier Schritte eingeteilt werden.

Reduktion: Das Wissen wird auf das Wesentliche reduziert.

Strukturierung: Zeitgleich mit der Reduktion werden jeweils zwei Begriffe räumlich zueinander angeordnet.

Visualisierung: Die Visualisierung der beiden ersten Schritte. Dieser Schritt wird auch zur selben Zeit durchgeführt.

Elaboration: Die bisherige Concept Map wird in diesem Schritt verfeinert, durch z.B. Beschriftung der Pfeile.

2. Grundlagen

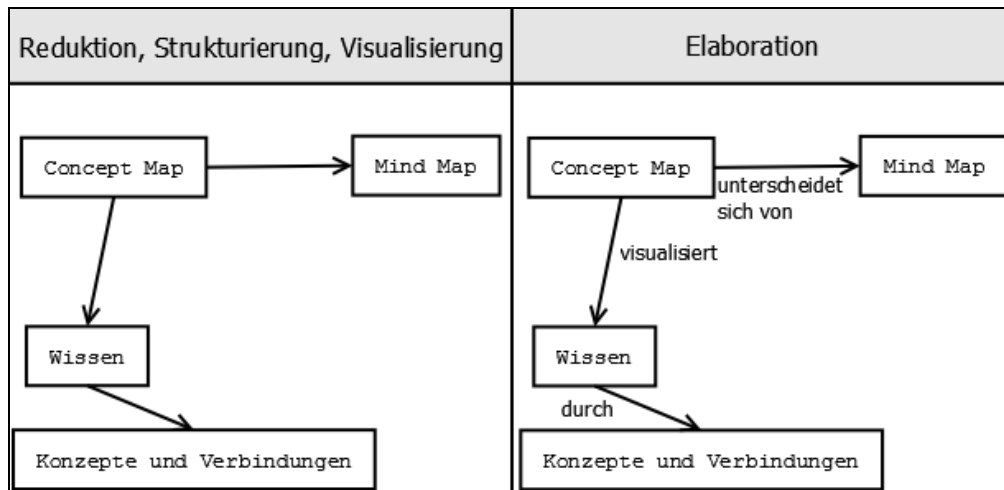


Abbildung 1: Erstellung von Concept Maps

Dargestellt sind in Abbildung 1 die Schritte zur Erstellung einer Concept Map. Im ersten Schritt wird das Darzustellende auf Begriffe konzentriert, räumlich zueinander angeordnet und schließlich dargestellt. Im zweiten Schritt wird die bisherige Darstellung verfeinert.

Mit diesen Mitteln wird bei der Erstellung einer Concept Map nun versucht bestehendes Wissen, ähnlich der in den eigenen Gedanken vorliegenden Struktur, darzustellen. Wenn es also darum geht Ideen niederzuschreiben, geht es nicht darum den Verstand zu entleeren, sondern darum ihn zu rekonstruieren (Sharples, in Ostwald 1996). Concept Maps dienen somit der strukturierten Darstellung von bestehendem Wissen und sind ein Mittel der Gedankenordnung und Reflexion.

In der Anwendung helfen Concept Maps, durch die Möglichkeit der Wissensstrukturierung und die intensivere Beschäftigung mit einem Thema, die Behaltensleistung zu verbessern. Bestehendes Wissen wird also gefestigt und Wissenslücken sind einfacher zu erkennen (Gurlitt & Nückles, 2010). Concept Maps helfen dabei ein komplexes Thema zusammenhängend sich vorzustellen und nicht nur als eine Sammlung von einzelnen Fakten. Es hat sich weiterhin gezeigt, dass einstmals erfolglose Studenten durch die Erstellung von Concept Maps Aufgaben meistern konnten und so ein Gefühl von Kontrolle über ein Themengebiet erlangen (Bascones & Novak, 1985). Es wird erwartet, dass Concept Maps neben dem Aufbau einer Wissensstruktur das Übertragen von Informationen auf neue Situationen ermöglichen (Van Zele, et al., 2004).

2.1.2. Concept Maps und Mind Maps

Concept Maps und Mind Maps sind auf den ersten Blick zwei sehr ähnliche Konzepte und teilen sich einige Eigenschaften. Jedoch unterscheiden sich die beiden Ansätze in ihren Anwendungszwecken (Eppler, 2006). Den ersten Hinweis auf die konzeptionellen Unterschiede kann man bereits schon an der Darstellung der beiden Konzepte erkennen.

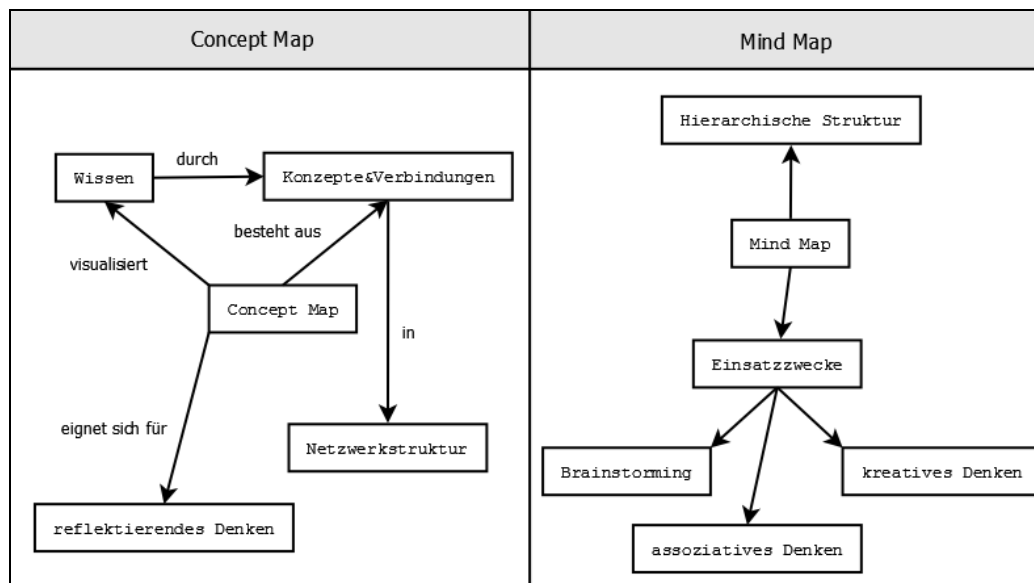


Abbildung 2 Concept Map und Mind Map

Concept Maps haben eine Netzwerkstruktur mit Verbindungen untereinander, während Mind Maps einer strengeren Hierarchie folgen. Oft gehen Verbindungen bei Mind Maps von einem zentralen Begriff aus, bei Concept Maps muss es so ein zentrales Konzept nicht geben.

So lassen sich auch die Unterschiede in der Anwendung der beiden Visualisierungsmöglichkeiten herleiten. Wie schon erwähnt, regen Concept Maps zum analysierenden und reflektierenden Denken an und ermöglichen die Visualisierung von bestehendem Wissen. Mind Maps regen vielmehr kreatives Denken an (Buzan 1974 in Eppler 2006) und dienen damit unter anderem als Brainstorming-Mittel. Die Art der Strukturierung und auch der eben angesprochene Prozess der Erstellung von Concept Maps kostet dabei mehr Zeit als die Erstellung von Mind Maps.

2.1.3. Kollaborative Concept Maps

Im Folgenden soll überprüft werden, ob das kollaborative Erstellen von Concept Maps einen Mehrgewinn für die individuellen Lernenden bietet. Dafür wird zunächst herausgestellt wie die gemeinsame Bearbeitung abläuft um daraufhin, in Bezug auf bestehende Studien, dieses zu bewerten.

Die Zusammenarbeit an Concept Maps kann am selben Ort an Tafeln oder Whiteboards, aber auch an unterschiedlichen Orten durch entsprechende Mehrbenutzereditoren erfolgen. Dabei möchte jeder Teilnehmer der Sitzung, ähnlich wie beim Prozess der individuellen Erstellung von Concept Maps, sein eigenes Wissen visualisieren.

Der grundlegende Prozess der Erstellung von Concept Maps bleibt somit zunächst der gleiche. Es wird allerdings nun in der endgültigen Concept Map nicht mehr nur das Wissen eines einzelnen Lernenden dargestellt, sondern dass der ganzen Gruppe. Kollaborative Concept Maps bieten also eine Basis für den Stand des gemeinsamen Wissens. Dies ist ein wichtiger Faktor bei der Erfassung von Wissen innerhalb von Gruppen (Ostwald, 1996).

2. Grundlagen

Generell wird bei der kollaborativen Arbeit an Concept Maps die synchrone Kommunikation der Asynchronen vorgezogen (De Simone, et al., 2001). Auch verändern sich andere Aspekte durch die gemeinsame Bearbeitung. So erleichtert zum Beispiel die gemeinsame Bearbeitung eines Problems die Aktivierung von bestehendem Wissen, genau das also, was durch Concept Maps dargestellt wird soll (Ostwald 1996, in Gao et al. 2007). Des Weiteren ist die Interaktion zwischen Lernenden eine wichtige Quelle für Ideen. Wenn die richtigen Fragen gestellt und Dialoge geführt werden, können vorgestellte Ideen andere Teilnehmer dazu anregen, diese weiter auszuarbeiten (Gunawardena, et al., 1997). Ein weiterer wichtiger Aspekt, der durch die gemeinsame Bearbeitung von Concept Maps hinzukommt ist, dass die individuellen Lernenden sich bei der gemeinsamen Bearbeitung einer Aktivität mehr engagieren müssen, um diese erfolgreich abzuschließen.

Im Laufe der Bearbeitung kollaborativer Concept Maps vertieft sich nicht nur das Verständnis der bisherigen Konzepte sondern es entwickeln sich daraus weitere neue Konzepte (Derry, et al., 2000, in Gao et al. 2007).

2.2. Kooperatives Lernen

Kooperatives Lernen ist ein Lernansatz, bei dem Lernende in Gruppen eingeteilt und so gemeinsam, computer-gestützt oder nicht, eine Aktivität bearbeiten. Bei so einer Lernaktivität soll das eigene Lernvermögen, sowie das der anderen Gruppenmitglieder maximiert werden (Johnson, et al., 1994). Johnson et al. (1991) beschreiben fünf sogenannte Säulen, auf denen das *Kooperative Lernen* aufbaut.

Positive interdependence oder auch positive Wechselbeziehung bezeichnet die Vorstellung jedes Individuums, dass die Arbeit mit anderen Studenten einen Wert hat.

Promotive, face-to-face interaction meint die Erwartung, dass Studenten sich bei der Zusammenarbeit gegenseitig helfen.

Als dritter wichtiger Aspekt wird *Individual accountability* vorgestellt, was bedeutet, dass die Leistung der Gruppenteilnehmer individuell erfasst werden kann. Das soll den Zweck haben, dass Jeder Verantwortung für seine Aufgaben in einer Gruppe übernimmt.

Social skills, die Fähigkeit mit anderen Studierenden zu interagieren sowie *group processing*, die Reflektion der Leistung der Gruppen durch die Mitglieder, stellen die letzten beiden vorgestellten Eckpfeiler für *Kooperatives Lernen* dar.

2.3. Computer Supported Collaborative Learning (CSCL)

2.3.1. Einführung

Computer-Supported-Collaborative-Learning ist das wissenschaftliche Feld welches sich damit beschäftigt, wie kollaboratives Lernen durch Computer unterstützt werden können. *Computer-Supported-Cooperative-Work* (CSCW) ist dagegen das Feld welches sich allgemein mit der computer-gestützten kollaborativen Zusammenarbeit an Aktivitäten beschäftigt (Howard 1989, in Eseryel et al. 2002).

Eingeführt wurde dieser Begriff 1984 von Greif und Cashman im Rahmen eines Workshops (Grudin, 1994).

Bei CSCL steht das zweite C nicht nur für *collaborative* sondern kann auch *cooperative*, *collective* oder *coordinated* bedeuten (Lipponen, 2002). In der Regel kann man die Begriffe Kooperation und Kollaboration danach unterscheiden, wie sehr die Arbeit unter Teilnehmern aufgeteilt wird. Kooperation bezeichnet meist die Aufteilung einer Aufgabe in Teilaufgaben, welche individuell bearbeitet werden. Kollaboration dagegen meint oft die gemeinsame Bearbeitung derselben Aufgabe (Kozar, 2010). Begriffe und Herausforderungen aus der CSCL und CSCW sollen im Folgenden vorgestellt werden.

2.3.2. Groupware

Bei Groupware handelt es sich um Software, welche Prinzipien der CSCW bzw. CSCL umsetzen. Also die gemeinsame Arbeit an Projekten mit Hilfe von Computern ermöglicht. Bei der Entwicklung von Groupware muss man sich einigen Herausforderungen stellen, wie zum Beispiel Awareness bei der Kollaboration zwischen Nutzern zu ermöglichen. In dieser direkten Interaktion unterscheidet sich Groupware auch am meisten von anderer Software, welche mehrere Nutzer unterstützen. (Lynch, et al., 1990).

2.3.3. Mehrbenutzereditoren

Mehrbenutzereditoren erlauben kollaborative Bearbeitung durch Synchronisierung des Dargestellten zwischen allen Teilnehmern. Die Bearbeitung der Aktivität durch Lernende kann hier in der Regel in synchrones sowie asynchrones Lernen unterteilt werden.

Synchrone Kommunikation bedeutet die Bearbeitung zur gleichen Zeit, während asynchron die Bearbeitung zu unterschiedlichen Zeitpunkten meint. Gerade Mehrbenutzereditoren versuchen in der Regel beide Fälle zu ermöglichen. So haben beide auch ihre eigenen Vorteile. Synchrones Lernen ermöglicht Interaktion sowie Kollaboration durch Lernende, diese beiden Punkte spielen einen wichtigen Faktor bei erfolgreichem Lernen (Bonk & Zhang, 2006, in Higley 2013).

Asynchrones Lernen dagegen ermöglicht es Studenten gemeinsam eine Aufgabe zu bearbeiten und dennoch nach ihrer eigenen Geschwindigkeit vorgehen um so Lernziele besser überdenken zu können (Garrison & Kanuka, 2004, in Higley 2013).

	Gleiche Zeit (Synchon)	Unterschiedliche Zeit (Asynchron)
Gleicher Ort	Präsentationssysteme	Projektmanagementsysteme
Unterschiedlicher Ort	Mehrbenutzereditoren	E-Mail

Abbildung 3 Time/Space Matrix

2. Grundlagen

2.3.4. Sitzung

Eine *Sitzung* kann man als den Zeitabschnitt betrachten in dem mehrere Benutzer, sogenannte Teilnehmer, gemeinsam synchron eine Aktivität durchführen. Die Erstellung und Verwaltung einer solchen Sitzung stellt im Rahmen der computer-gestützten Zusammenarbeit eine Herausforderung dar.

2.3.5. Geteilter Kontext

Unter anderem im Fall von Mehrbenutzereditoren teilen sich die Teilnehmer einer Sitzung gewisse Komponenten der Oberfläche. Diese Komponenten nennt man auch *Geteilter Kontext*. Neben der Awareness ist dieser Begriff ein weiterer zentraler Aspekt bei Groupware. In der weiteren Arbeit wird der Begriff Darstellung synonym mit dem Begriff des geteilten Kontexts verwendet.

2.3.6. Herausforderungen

2.3.6.1. Einführung

Bei der kollaborativen Zusammenarbeit an unterschiedlichen Orten gibt es Herausforderungen welche bei der Bearbeitung von Aktivitäten am selben Ort auf natürliche Weise geregelt werden oder erst gar kein Problem darstellen. Folgende Herausforderungen müssen somit bei der Entwicklung von Groupware in Betracht gezogen werden.

2.3.6.2. Awareness

Awareness ist ein zentraler Begriff und wichtiger Punkt der bei der Entwicklung von Groupware zu beachten ist. Kollaborative Bearbeitung von Aufgaben an gleichem Ort ermöglicht direktes Feedback aller Teilnehmer, die Kommunikation, Absprache und Koordination geschieht auf natürliche Weise. Ohne diese natürliche Interaktion fühlt sich die Zusammenarbeit in Groupware System oft ineffizient und unbeholfen an (Gutwin & Greenberg, 2002). Diese wichtigen Mittel zum erfolgreichen Bearbeiten einer Aufgabe müssen im Kontext der computer-gestützten Zusammenarbeit an unterschiedlichen Orten durch Software ermöglicht werden. Interaktion zwischen Teilnehmern soll damit möglichst natürlich stattfinden können und es soll ein Gefühl von Awareness über die anderen Teilnehmer einer Aktivität aufkommen. Dies kann erreicht werden durch eine möglichst direkte Synchronisation aller Aktionen, durch Wahrung der Konsistenz, durch Kenntnis machen über die Aktionen der Anderen, sowie durch Mittel zur Kommunikation.

2.3.6.3. Throughput / Direktheit

Diese Begriffe beschreiben wie gut bzw. zeitnah Aktionen anderer Teilnehmer in einer kollaborativen Sitzung sichtbar sind. Wie zeitnah zum Beispiel in einem Mehrbenutzer-Texteditor die Tastatureingaben auch bei anderen Teilnehmern erscheinen, lassen sich mit diesen Begriffen beschreiben.

2.3.6.4. Nebenläufigkeit

Bei kollaborativen Anwendungen kann es durch Benutzeraktionen der Teilnehmer zu konkurrierenden Befehlen bzw. Aktionen kommen. Solche Anwendungen müssen die Nebenläufigkeit zeitgleicher Aktionen ermöglichen.

2.3.6.5. Persistenz

In Mehrbenutzereditoren muss die gemeinsam bearbeitete Ressource bzw. das Dokument auf irgendeine Art und Weise gespeichert werden. Dies kann auf der Seite der Clients oder zentral auf einem Server geschehen. Um die asynchrone Bearbeitung der Ressource zu ermöglichen ist es notwendig die Ressource an einer zentralen Stelle zu speichern, sodass jeder Teilnehmer darauf zu beliebigen Zeitpunkten darauf zugreifen kann.

2.4. Architekturen

Bei der Entwicklung von Anwendungen, welche kollaboratives Arbeiten ermöglichen sollen, kann man sich nach unterschiedlichen Architekturen richten. Solche Architekturen müssen im Fall von Mehrbenutzereditoren insbesondere die synchrone und asynchrone Bearbeitung ermöglichen. Die hier vorgestellten Architekturen bieten alle Vor- und Nach-/Teile in Bezug auf Coupling bzw. Direktheit, Integrierbarkeit und Netzwerklast.

2.4.1. Coupling

Suthers (2001) beschreibt das Prinzip von Coupling, welches die Kupplung zwischen Benutzeraktionen und den Zuständen der Anwendung darstellt. Weiterhin werden drei Kategorien für den Grad dieser Kupplung definiert.

Strict: What you see is what I see (WYSIWIS):

Hier haben alle Nutzer stets dieselbe View sowie Controller Zustände. Die Begriffe View und Controller bezeichnen hierbei die gleichnamigen Begriffe aus dem MVC-design pattern. Aktivitäten der Nutzer sind hier eng gekuppelt.

Relaxed WYSIWIS:

Nicht alle Ansichten müssen ganz genau dieselben sein. Man kann z.B. im gleichen Fenster scrollen, ohne dass diese Aktivität synchronisiert wird.

2. Grundlagen

Model Level coupling:

What you model is what I model (WYMIWIM). Hier wird nur garantiert dass alle Nutzer die exakt gleiche Semantik des Modells sehen. Die Ansichten können prinzipiell komplett unterschiedlich sein. So können zum Beispiel ganz andere Darstellungen bei allen Nutzern für dasselbe Model gegeben sein .

2.4.2. Zentrale Architektur

In einer zentralen Architektur gibt es nur eine zentrale Anwendung. Es werden die Darstellung sowie Möglichkeiten zur Veränderung des zentralen Modells an alle Teilnehmer verteilt. Das System des Nutzers ist in diesem Fall als *Thin Client* zu sehen, nur die Darstellung und Übergabe von Befehlen erfolgt hier. So werden bei dieser Architektur die Benutzeraktionen der Clients an die zentrale Stelle weitergeleitet, welche dann die Ausgabe an alle Clients sendet. Alle Berechnungen und Aktualisierungen des Modells erfolgen serverseitig. Der große Vorteil dieser Implementation ist, dass jede beliebige Anwendung kollaborativ umgesetzt werden kann. Auch kann die Konsistenz der Darstellungen der Teilnehmer auf diese Weise leicht gesichert werden. Eine zentralisierte Architektur bedeutet in der Regel eine enge Kupplung zwischen den Benutzeraktionen der Teilnehmer in einer kollaborativen Sitzung bis hin zu einem unerwünschten Grad. Ein weiterer Nachteil ist, dass durch die Übertragung der kompletten Darstellung eine hohe Bandbreite anfällt. Weiterhin sind Datenstau und hohe Antwortzeiten ein Problem (Balter, et al., 1995).

2.4.3. Replizierte Architektur

In einer replizierten Architektur befinden sich die Anwendungen vollständig auf den Systemen der Teilnehmer. Im Vergleich zur zentralen Architektur ist die Netzwerk-Auslastung hier verbessert. Nur Steuerbefehle werden über das Netzwerk vermittelt. Gegenüber der zentralen Architektur erlaubt die replizierte Architektur eine weniger direkte Kupplung der Benutzeraktionen unter den Teilnehmern. So spricht man mehr die Lernenden an, welche hauptsächlich an den semantischen Änderungen des Modells interessiert sind (Suthers, 2001). Auch andere Nachteile der zentralen Architektur, wie zum Beispiel die hohen Antwortzeiten und generell eine höhere Robustheit sind hier nicht mehr vorhanden (Balter, et al., 1995).

2.4.4. Verteilte Architektur

Die verteilte Architektur unterscheidet sich von der Zentralen insofern, dass hier in der Regel nur das Modell serverseitig vorliegt. Clients führen Operationen auf diesem Modell aus und stellen es lokal dar. In diesem Fall müssen nur Updates des Modells an den Server geschickt werden. Der Zugriff auf dieses Modell erlaubt daraufhin die Darstellung der am Modell vorgenommenen Änderungen. Dieser Ansatz ist sehr schonend in Bezug auf Netzwerkressourcen, erfordert jedoch eine ständige Verbindung (Suthers, 2001).

2.4.5. Hybride Architektur

Hybride Ansätze können Elemente aus der Verteilten sowie Replizierten Architektur enthalten. Zum Beispiel kann die Persistenz über ein zentral gespeichertes Modell gesichert werden, also Aspekte der verteilten Architektur beinhalten. View und Controller liegen hier repliziert auf den Systemen der jeweiligen Teilnehmer vor. Nachrichten an das zentralisierte Modell erlauben die Synchronisation.

2.5. Moodle

Moodle³ ist eine modulare Lernplattform. Fast alle grundlegenden und erweiterten Funktionen werden in Moodle über so genannte Module bereitgestellt. Module gehören dabei immer zu einem bestimmten Typ, nämlich einer Aktivität oder Ressource.

Aktivitäten können instanziiert werden und anschließend können Studierende darauf zugreifen, um diese alleine oder in Interaktion mit Lehrenden bzw. anderen Studenten zu bearbeiten. Moodle bietet weitere Funktionalitäten, wie zum Beispiel die Möglichkeit Lehrende zu Gruppen zuzuordnen. Aktivitäten können so zum Beispiel Kollaboration bzw. Kooperation innerhalb dieser Gruppen ermöglichen. Gruppen in Moodle können auf zwei Levels benutzt werden. Es kann der Gruppenmodus für den gesamten Kurs und damit alle darin befindlichen Aktivitäten bestimmt werden. Einstellungen zu der Benutzung von Moodle Gruppen können auch individuell für jede Aktivität einzeln vorgenommen werden. Dabei umfassen diese Einstellungen die Festlegung, ob keine Rücksicht auf Gruppen genommen werden soll (*no groups*), ob Gruppen nur ihre eigene Gruppe sehen können (*separate groups*), oder aber ob Gruppen auch die Arbeit von anderen Gruppen sehen können (*visible groups*).

2.6. TogetherJS

Die JavaScript Bibliothek TogetherJS⁴ bietet kollaborative Funktionen und Werkzeuge zur Erweiterung einer Website. Mehrere Nutzer können über eine geteilte URL gemeinsam eine Ressource betrachten. In einer so aufgebauten Sitzung werden die aktuelle Position der Mauszeiger und Klicks aller Nutzer für alle kenntlich gemacht. Informationen über die Teilnehmer in einer Sitzung werden in einer Leiste an der Seite des Bildschirms angezeigt. Hier besteht auch die Möglichkeit der Kommunikation zwischen den Teilnehmer mittels Text Chat sowie Sprach-Chat über WebRTC.

Die wichtigste Komponente von TogetherJS ist der so genannte *hub-Server*, über den die gesamte Kommunikation läuft. Die Sitzung wird über den hub-Server geregelt, so gehen alle Aktionen von Teilnehmer einer Sitzung als Nachricht zunächst an den hub-Server, welche diese an alle Teilnehmer der Sitzung weiterleitet. TogetherJS bietet weiterhin eine einfache Schnittstelle um über WebSockets und den hub-Server, personalisierte Nachrichten an alle Sitzungsteilnehmer zu versenden. So können viele Anwendungen um Kollaboration erweitert werden.

³ <https://moodle.org/>

⁴ <https://togetherjs.com/>

2.7. Go-Lab

Das Go-Lab Projekt⁵ ist ein von der Europäischen Kommission, im Rahmen des Siebten Rahmenprogramms, teil-finanziertes Projekt. Go-Lab steht dabei für Global Online Science Labs for Inquiry Learning at School. Beteiligt sind 19 Organisationen aus 12 Ländern.

Go-Lab bietet den Zugriff auf sogenannte Online Labore an. Der Zugriff auf diese Labore ist weltweit über das Go-Lab Portal möglich. Die angebotenen Labore ermöglichen es Lernenden durch die Durchführung von Experimenten ihr Wissen zu vertiefen (Go-Lab Project, 2014).

Neben Laboren werden Lernenden Werkzeuge zur Lernunterstützung zur Verfügung gestellt. Dazu gehört auch das in dieser Arbeit verwendete Concept Mapping Tool.

⁵ <http://www.go-lab-project.eu>

3. Ansatz

In diesem Kapitel werden zunächst die Anforderungen an die kollaborative Bearbeitung von Concept Maps herausgearbeitet. Ebenfalls werden die Anforderungen, die im Kontext von Moodle und bei der Entwicklung eines Moodle Plug-Ins entstehen, formuliert. Anschließend wird dann die Architektur der gesamten Anwendung, sowie einzelner Komponenten, vorgestellt. Die hier herausgearbeiteten Funktionen und Komponenten stellen den Ansatz dieser Arbeit dar.

3.1. Einführung

Das zu entwickelnde Plug-In baut, wie schon in den Grundlagen bereits erwähnt, auf einem bestehendem Concept Mapping Tool auf. Mit dieser Anwendung sind alle grundlegenden Funktionen zur Erstellung einer Concept Map gegeben. Innerhalb dieser Arbeit soll diese Anwendung um kollaborative Funktionalitäten erweitert werden. Weiterhin soll aus dieser Anwendung ein Plug-In für Moodle entwickelt werden.

Studenten müssen in einer Sitzung gemeinsam eine Concept Map bearbeiten können. Wie im Folgenden vorgestellt wird, muss dazu die synchrone sowie auch die asynchrone Bearbeitung ermöglicht werden. Des Weiteren müssen im Rahmen von Moodle neue Features eingeführt werden, wie eine Konfigurierbarkeit des Plug-Ins, die Möglichkeit Abgaben zu kommentieren und Protokolle von Benutzeraktionen einzusehen.

3.2. Anforderungen an die Synchronisierung

Wie bereits angesprochen, muss für die kollaborative Zusammenarbeit die synchrone Bearbeitung zwischen allen Teilnehmern einer Sitzung ermöglicht werden. Es werden im Folgenden alle nötigen Schritte vorgestellt, um die Synchronisierung zu sichern. So werden auch die in der Anwendung gewünschten Funktionen vorgeführt.

3.2.1. Synchronisierung von Benutzeraktionen

Bei der Bearbeitung von Concept Maps führen Benutzer Aktionen durch, wie zum Beispiel das Erstellen eines Konzeptes. Solch eine Benutzeraktion wird lokal in der Anwendung entsprechend ein neues Konzept darstellen. Um die Synchronisierung dieser Aktionen über alle Teilnehmer einer kollaborativen Sitzung hinweg zu ermöglichen, werden bei der Auslösung einer Aktion zusätzlich Nachrichten an die Teilnehmer verschickt. Der Empfänger einer solchen Nachricht kann mit den erhaltenen Informationen die Aktion lokal wiederholen.

3. Ansatz

Dazu müssen alle nötigen Informationen zur Wiederholung der Aktion gesammelt und an die anderen Teilnehmer versandt werden. Dieser Nachrichtenversand geschieht mit Hilfe der JavaScript Bibliothek TogetherJS über WebSockets. Die genauere Architektur dieser Synchronisation wird später in diesem Kapitel noch einmal im Detail vorgestellt.

Werden Benutzeraktionen auf diese Weise übermittelt, so wird eine synchrone Bearbeitung der Concept Map möglich. Die Benutzer werden in diesem Ansatz sofort über Änderungen an den Concept Maps von anderen Teilnehmern informiert.

3.2.2. Synchronisierung neuer Teilnehmer

Bisher werden alle Aktionen der Teilnehmer einer kollaborativen Sitzung synchronisiert. Kommt jedoch ein neuer Teilnehmer hinzu, so reicht dieses Prinzip nicht mehr aus.

Neue Teilnehmer in der Sitzung müssen also auf den aktuellen Stand der Bearbeitung gebracht werden. Dies geschieht durch eine einmalige Synchronisierung der gesamten Concept Map bei der Initialisierung auf der Seite des neuen Teilnehmers. Dazu sendet dieser eine Nachricht, woraufhin alle anderen Teilnehmer mit dem aktuellen Stand ihrer Concept Map antworten. Die Concept Map wird vom neuen Teilnehmer geladen und kann nun synchron von allen Teilnehmern bearbeitet werden.

3.2.3. Wahrung der Konsistenz

Weiterhin müssen Fehler bei der Übertragung von Benutzeraktionen und konkurrierende Aktionen von Benutzern beachtet werden. Diese können zu unterschiedlichen Darstellungen der Maps bei den Teilnehmern einer Sitzung führen. Anstatt solche Fehler zu beheben, soll in diesem Ansatz die Behandlung nach dem Auftreten solcher Fehler durchgeführt werden. Dazu wird von jedem Teilnehmer in gewissen Intervallen die eigene Concept Map an alle anderen Teilnehmer der kollaborativen Sitzung gesendet. Auf diese Weise wird stets sichergestellt, dass, sollte ein Unterschied in der Darstellung vorkommen, alle Teilnehmer wieder dieselbe Concept Map sehen. Es verändert sich jedoch für Einen oder Mehrere die eigene Concept Map.

3.2.4. Zusammenfassung

Mit den eben vorgestellten Funktionen ist es möglich den bestehenden Concept Mapper so zu erweitern, dass Teilnehmer synchron Concept Maps innerhalb einer kollaborativen Sitzung bearbeiten können.

Die Art des hier beschriebenen Umgangs mit der Synchronisierung ermöglicht direktes Feedback über Veränderungen an der Concept Map durch andere Teilnehmer. Ebenso gibt es mit Hilfe von TogetherJS Feedback darüber, wer an der Sitzung teilnimmt und wo sie sich in räumlich im Editor befinden. Sprach- sowie Text-/Chat erlaubt ermöglicht Kommunikation und Absprache unter den Lernenden. Die Gesamtheit dieser Ansätze soll so auch das Gefühl von Awareness bei der Kollaboration gewährleisten.

3.3. Anforderungen an das Plug-In

3.3.1. Einführung

Im Kontext von Moodle muss davon ausgegangen werden, dass Studenten nicht immer zur gleichen Zeit an einer Concept Map arbeiten werden. So muss in diesem Zusammenhang also auch die asynchrone Bearbeitung möglich gemacht werden. Im Folgenden wird diese und weitere Anforderungen, die, neben der direkten Synchronisierung, an das Plug-In gestellt werden können, formuliert.

3.3.2. Sitzungsmanagement

TogetherJS ermöglicht den Aufbau einer Sitzung. Normalerweise geschieht dies über einen zwischen allen Teilnehmern geteilten Link. Im Rahmen des Plug-Ins soll jedoch die Sitzungsverwaltung intern gehandhabt werden. Dazu wird bei Bedarf ein Schlüssel generiert und beim Start des Plug-Ins an TogetherJS weitergereicht.

3.3.3. Asynchrone Bearbeitung

Um die asynchrone Bearbeitung zu ermöglichen wird die Concept Map zentral in der Moodle Datenbank gespeichert. So wird, auch durch regelmäßige Speicherung bei der Bearbeitung, die Persistenz der Concept Maps gesichert. Die zentrale Speicherung auf dem Server hat den Vorteil, dass nur eine Kopie der Concept Map gesichert werden muss, auch ist mit diesem Ansatz die gesamte Architektur der Anwendung simpler.

3.3.4. Unterstützung von Moodle Gruppen

Moodle bietet die Möglichkeit einzelne Lernende zu Gruppen zuzuweisen. Im Plug-In soll eingestellt werden können, ob diese Gruppen beachtet werden. Im Gruppenmodus sieht jede Gruppe für jede Instanz des Plug-Ins ihre eigene Concept Map. Ist der Gruppenmodus nicht ausgewählt, arbeiten für jede Instanz des Plug-Ins alle Studenten in einem Kurs zusammen an derselben Concept Map.

3. Ansatz

3.3.5. Einstellungsmöglichkeiten

Das zu entwickelnde Plug-In soll bei seiner Instanziierung folgende Einstellungsmöglichkeiten erlauben:

- Bestimmung des Namens
- Einstellung des Bearbeitungszeitraums
- Einstellung des Gruppenmodus

Zu den grundlegenden Funktionen die ein Moodle Plug-In bieten muss, gehört, neben diesen Einstellungsmöglichkeiten, auch ganz allgemein das Erstellen und Löschen des Plug-Ins. So muss beim Löschen des hier zu entwickelnden Plug-Ins nicht nur die Aktivität selbst gelöscht werden. Es müssen auch die, innerhalb einer Aktivität erstellten, Concept Maps und Protokolle gelöscht werden.

3.3.6. Protokollierung

Das Plug-In soll wichtige Benutzeraktionen während der kollaborativen Bearbeitung der Maps festhalten. Es soll bei Aktionen, wie zum Beispiel dem Hinzufügen von Verbindungen, unter anderem der Typ der Aktion, Name des Akteurs, sowie seine Gruppe nachgehalten werden. Diese Informationen werden in der Moodle Datenbank gespeichert und der Lehrperson in Form einer Textausgabe zur Verfügung gestellt.

3.3.7. Kommentarfunktion

Lehrende müssen die Möglichkeit haben, Kommentare zu den Leistungen der einzelnen Studenten bzw. Gruppen abgeben zu können. Die Abgabe von Kommentaren wird nach Ablauf der Bearbeitungszeit ermöglicht. Kommentare können nach deren Abgabe von den Studenten eingesehen werden.

3.3.8. Darstellung

Bisher wurde noch nicht beachtet, dass Lernende und Lehrende andere Anforderungen an die Funktionalität des Plug-Ins haben. Hierzu müssen Darstellung sowie Funktionalität generell danach unterschieden werden, ob sich die Aktivität innerhalb des Bearbeitungszeitraums befindet oder außerhalb. Im Folgenden sollen in kurzen Anwendungsfällen die unterschiedlichen Sichten der Benutzergruppen vorgestellt und weiter noch alle bisher genannten Funktionen in einem praktischen Bezug dargestellt werden.

Generell kann man zwei Nutzergruppen für die Anwendung definieren. Diese werden in den folgenden Szenarien als Studenten und Lehrende beschrieben. Des Weiteren kann man das Verhalten der Anwendung grundsätzlich danach aufteilen, ob man sich innerhalb oder außerhalb des Bearbeitungszeitraums befindet.

Im Folgenden soll für diese vier Fälle jeweils ein kurzer Anwendungsfall beschrieben werden. Die beschriebenen Zugriffsrechte sind im Plug-In als Capabilities definiert.

3.3.8.1. Student - Innerhalb des Bearbeitungszeitraums

Studenten können im Kurs eine Concept Mapper Aktivität aufrufen.

Befindet sich die Aktivität im Gruppenmodus wird nun für jede Gruppe eine neue Concept Map in der Datenbank von Moodle erstellt. Ist die Aktivität nicht im Gruppenmodus, so bearbeiten alle Studenten in dieser Instanz des Plug-Ins dieselbe Concept Map. Es wird die Möglichkeit von Moodle genutzt, Lehrende in Gruppen einteilen zu können.

Nachdem der Student das Plug-In geöffnet hat, soll er das Concept Mapping Tool angezeigt bekommen und die kollaborative Sitzung wird gestartet. Nun können Konzepte und Verbindungen erstellt und gelöscht werden, um so die Concept Map zu erstellen.

Kommt nach einer gewissen Zeit ein neuer Teilnehmer hinzu, überträgt der erste Student seinen aktuellen Stand der Bearbeitung an den neuen Teilnehmer. Ab diesem Moment haben beide Studenten in der kollaborativen Sitzung dieselbe Darstellung. Um diese Synchronität so zu erhalten, werden im weiteren Verlauf alle wichtigen Aktionen der Benutzer wie bereits beschrieben synchronisiert. Diese Benutzeraktionen werden ebenfalls protokolliert.

Die regelmäßige Synchronisierung der Concept Maps aller Teilnehmer sorgt während der Bearbeitung für die Konsistenz in der Sitzung.

Die Concept Map wird in Intervallen automatisch in der Moodle Datenbank gespeichert. Auch kann der Student die Concept Map zu jeder Zeit manuell speichern. Da immer nur eine Concept Map pro Instanz und Gruppe bearbeitet wird, wird diese nach Ablauf der Bearbeitungszeit als die Abgabe des Studenten interpretiert.

3.3.8.2. Lehrende - Innerhalb des Bearbeitungszeitraums

Lehrende können bei dem Hinzufügen der Aktivität die beschriebenen Einstellungsmöglichkeiten vornehmen. Wird nun die erstellte Aktivität aufgerufen, wird im Gruppenmodus eine Übersicht über alle bisher von Studenten erstellten Concept Maps angezeigt.

Der Lehrende kann nun eine davon auswählen, woraufhin eine nicht bearbeitbare Version des Concept Mapping Tools angezeigt wird. Ebenfalls erscheint die chronologische Protokollierung der bisher ausgeführten Benutzeraktionen bei der Erstellung der angezeigten Concept Map. Ist der Gruppenmodus nicht ausgewählt worden, so wird direkt die eine gemeinsame Concept Map angezeigt, ebenfalls nicht bearbeitbar und wieder mit der Einsicht auf die protokollierten Benutzeraktionen.

3. Ansatz

3.3.8.3. Lehrende - Außerhalb des Bearbeitungszeitraums

Hier erscheint wie im gerade eben beschriebenen Fall im Gruppenmodus zunächst der Auswahlbildschirm. In der folgenden Ansicht wird dem Lehrenden neben der nicht bearbeitbaren Concept Map und dem Protokoll der Benutzeraktionen, nun zusätzlich ein Kommentarfeld angezeigt. Hier kann ein Kommentar zu der abgegebenen Arbeit verfasst und gespeichert werden.

3.3.8.4. Student - Außerhalb des Bearbeitungszeitraums

Studenten können außerhalb des Bearbeitungszeitraums die Aktivität weiterhin aufrufen. Ihnen wird jedoch nun, ebenso wie der Lehrperson, eine nicht bearbeitbare Version der Concept Map angezeigt. Zusätzlich können Studenten die von Lehrenden abgegebenen Kommentare einsehen.

3.4. Zusammenfassung aller Anforderungen

Neben der Entwicklung des eigentlichen Plug-Ins können wir die folgenden Funktionen formulieren, die bei der Implementierung umgesetzt werden sollen. Damit sollen alle an das Plug-In gestellten Anforderungen ermöglicht werden.

- Speicherung von Concept Maps in der Moodle Datenbank
- Konfigurierbarkeit des Plug-Ins (Name, Gruppenmodus, Bearbeitungszeitraum)
- Verschiedene Sichten für die unterschiedlichen Nutzer des Plug-Ins
- Kommentarfunktion für Lehrende
- Protokollierung sowie Ausgabe von Benutzeraktionen
- Synchronisierung von Benutzeraktionen
- Behandlung von neuen Teilnehmern in einer Sitzung
- Umgang mit Übertragungsfehlern und konkurrierenden Aktionen

3.5. Architektur des Plug-Ins

Nachdem nun alle Funktionen herausgearbeitet und vorgestellt wurden, soll in diesem Abschnitt die Architektur des Plug-Ins vorgestellt werden. Damit wird zum einen der Aufbau des Plug-Ins dargestellt und ebenso noch einmal ein Gesamtüberblick über die wichtigsten Funktionen der Anwendung geboten. Es werden die einzelnen Elemente des Plug-Ins vorgestellt und deren Beziehung untereinander dargestellt. Details über die genaue Implementierung folgen im nächsten Kapitel.

Das Projekt baut auf einem bestehenden Programm auf, welches für sich genommen die Möglichkeit bietet in Einzelarbeit Concept Maps zu erstellen. Dieser Concept Mapper soll im Rahmen dieser Arbeit so erweitert werden, dass die Erstellung von Concept Maps auch in Gruppenarbeit möglich ist.

Darauf aufbauend gilt es ein Moodle Plug-In zu entwickeln, sodass die kollaborative Erstellung von Concept Maps im Rahmen von Moodle ermöglicht wird.

Es wird im Folgenden die grundlegende Architektur des Plug-Ins vorgestellt, und im Anschluss schematisch die Synchronisierung mit Hilfe von TogetherJS verdeutlicht.

3.5.1. Überblick

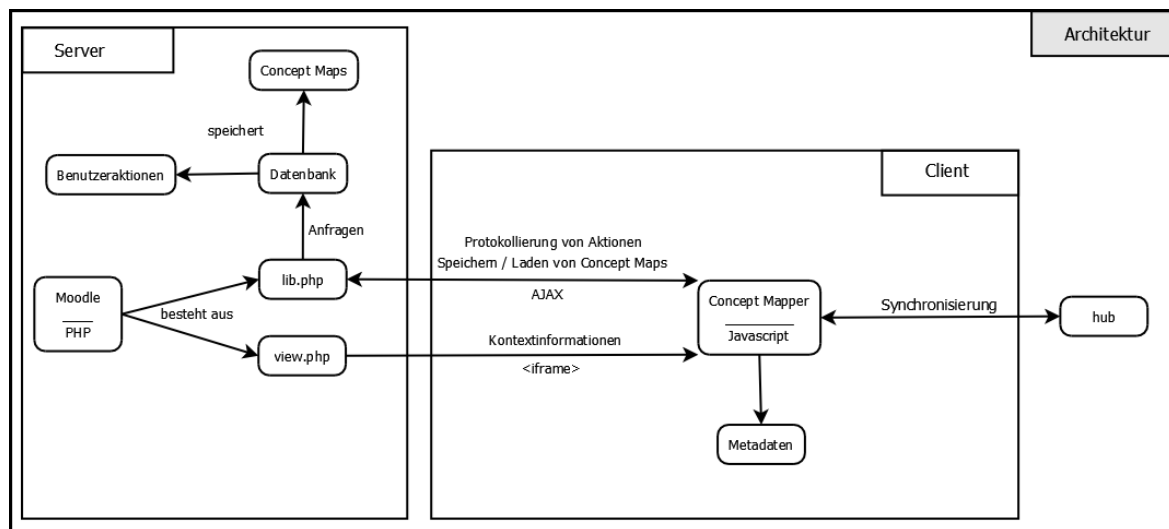


Abbildung 4 Architektur des Plug-Ins

Zu sehen ist in Abbildung 4 ein grober Überblick über die Komponenten des Plug-Ins. Zu bemerken ist, dass das Concept Mapping Tool mit Hilfe eines Inlineframes eingebettet wurde. Somit läuft ohne weitere Anpassung der Concept Mapper ohne jede Informationen über das Plug-In und Nutzer. Das ist auch der Grund warum in dieser Arbeit die Anpassung des Concept Mappers und die Entwicklung des Plug-Ins meist voneinander unterschieden werden. Die einzige Möglichkeit dem Concept Mapper Kontextinformationen zu übergeben besteht beim Aufruf des *iframes* über die URL.

Zur Speicherung der Concept Maps ruft das Concept Mapping Tool auf der Seite des Clients mittels AJAX, in der *lib.php*, serverseitige Funktionen des Plug-Ins auf. Hier erfolgt letztendlich das Speichern und Laden der Concept Map in und aus der Moodle Datenbank. Auf dieselbe Weise werden wichtige Aktionen der Nutzer protokolliert. In der View des Plug-Ins wird die Logik für die Sichten realisiert. Hier wird bestimmt wer welche Anzeige erhält. Hier wird diese Anzeige dann auch entsprechend dargestellt. Weiterhin werden hier dem Concept Mapper alle wichtigen Kontextinformationen mitgegeben damit unter anderem das Laden und Speichern von Concept Maps funktionieren kann.

3. Ansatz

In Bezug auf die im Kapitel zu den Grundlagen vorgestellten Architekturen, kann man die hier vorgestellte Architektur als eine hybride Architektur einstufen. Client-seitig ist die Möglichkeit zur Erstellung und Darstellung von Concept Maps gegeben. Die auf dem Server instanziierte Aktivität bietet die Möglichkeit die Concept Maps zentral in der Moodle Datenbank zu speichern und Benutzeraktionen werden zwischen den Clients synchronisiert.

Mit diesem Aufbau lassen sich alle definierten Funktionen für das Plug-In umsetzen. Auf die genaue Implementierung der Funktionen wird im nächsten Kapitel eingegangen.

3.5.2. Synchronisierung mit TogetherJS

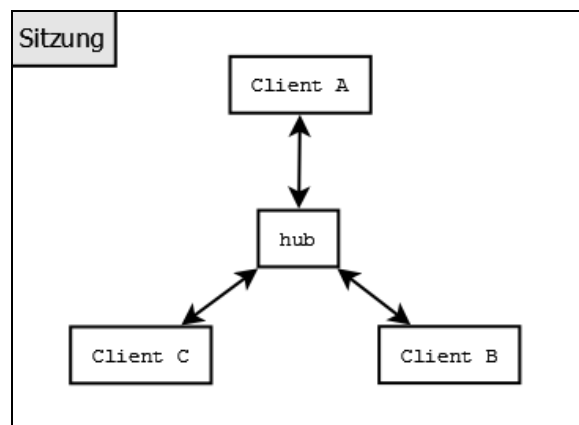


Abbildung 5 Synchronisierung über TogetherJS

Zu sehen ist in diesem Diagramm der grundlegende Aufbau der Kommunikation mittels TogetherJS. Der hub-Server ist hier klar als zentrales Element zu erkennen. Als Beispiel sind hier drei Teilnehmer in einer gemeinsamen Sitzung, dargestellt sind dabei deren Clients über die diese Kommunikation stattfindet. Über den hub-Server von TogetherJS findet der Aufbau einer Sitzung über einen Schlüssel, sowie der Beitritt zu einer Sitzung statt. Weiter werden über den hub Server die Nachrichten eines Sitzungsteilnehmers an alle anderen Teilnehmer weitergeleitet.

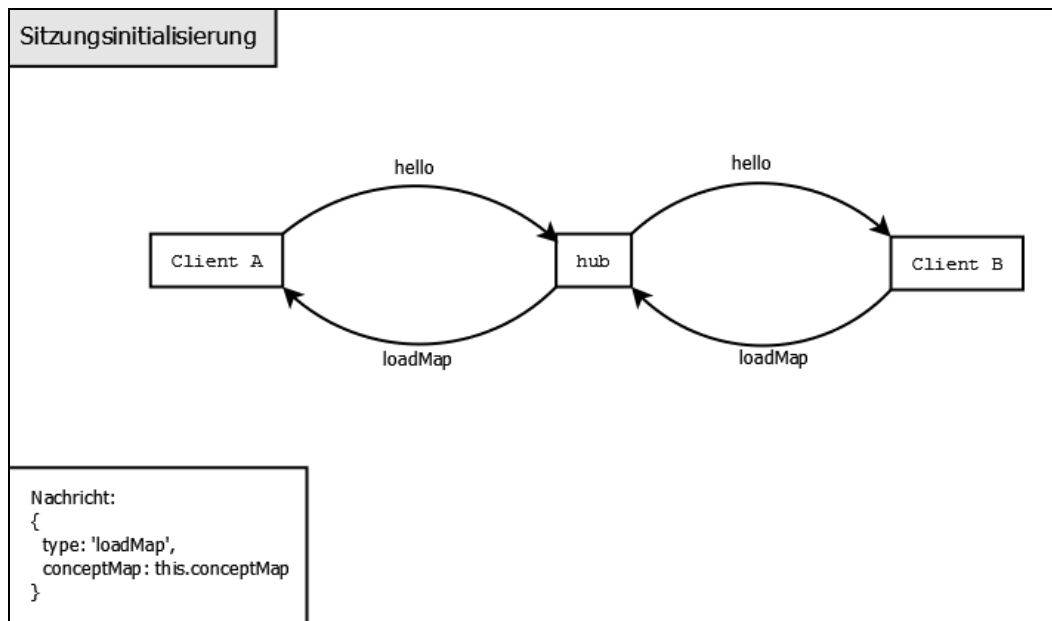


Abbildung 6 Nachrichtenaustausch über TogetherJS

In Abbildung 6 ist beispielhaft der Nachrichtenaustausch visualisiert, welcher für die Initialisierung der Concept Map eines neuen Sitzungsteilnehmers notwendig ist. Tritt also ein neuer Teilnehmer der Sitzung bei, so sendet dieser automatisch eine Nachricht an den hub Server.

Wie zu sehen ist, hat eine Nachricht immer einen Typ, welcher notwendig ist um die Nachricht zu interpretieren, und zusätzlich noch weitere Informationen. In diesem Fall werden als weitere Information die eigene Concept Map in der Nachricht mitgegeben.

Nachdem der hub Server die Nachricht an die anderen Teilnehmer verschickt hat, wird von diesen die eigene Concept Map wiederrum als Nachricht versendet. Dies erlaubt es dem neuen Teilnehmer die empfangene Concept Map darzustellen. Jede weitere wichtige Benutzeraktion wird auf dieselbe Weise unter allen Teilnehmer synchronisiert. TogetherJS hilft bei dieser Synchronisierung durch den hub-Server und den einfachen Nachrichtenaustausch. Die Anwendung muss an den entsprechenden Stellen angepasst werden um auf diese Art die Aktionen zu synchronisieren.

3. Ansatz

4. Implementierung

Dieses Kapitel soll die Implementierung der in Kapitel 3 herausgearbeiteten Funktionen erläutern. Dabei werden im Folgenden nur die umfangreicheren Umsetzungen vorgestellt. Auch werden Änderungen, welche an dem ursprünglichen Concept Mapper vorgenommen wurden, detaillierter beschrieben als die neuen Komponenten des Plug-Ins.

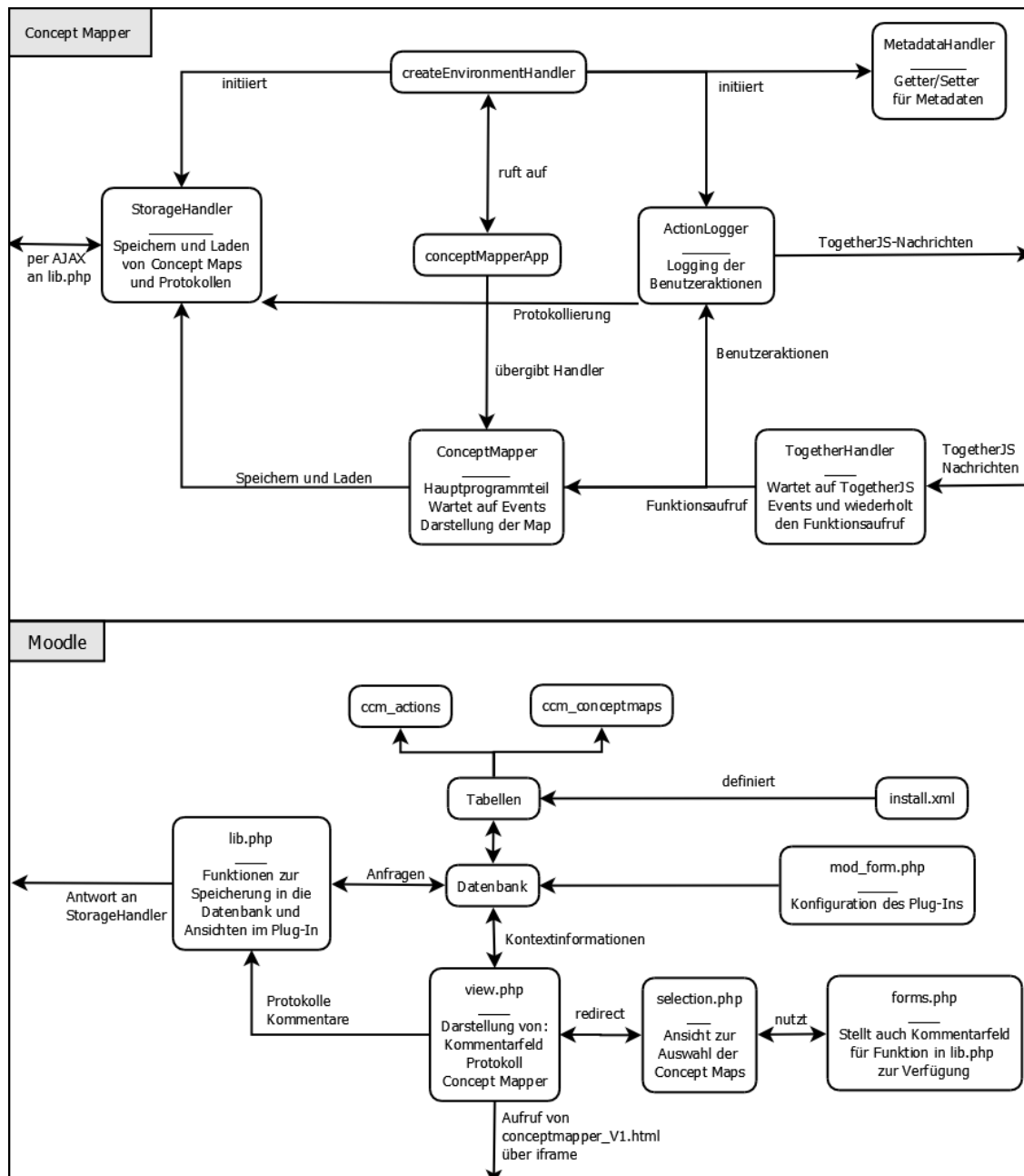


Abbildung 7 Detaillierter Aufbau des Plug-Ins

Dargestellt ist hier ein Gesamtüberblick über alle wichtigen Komponenten des Plug-Ins. Auch werden Funktionen der Komponenten kurz beschrieben und die Verbindungen der Komponenten innerhalb der Anwendung skizziert.

4. Implementierung

Der bestehende Concept Mapper ist in JavaScript implementiert und nutzt diverse Bibliotheken wie unter anderem JQuery, Angular und jsPlumb. Die Erweiterung innerhalb des Concept Mapping Tools, um Synchronisierung und Funktionen des Plug-Ins sind somit natürlich auch in JavaScript gehalten. Moodle dagegen nutzt PHP als Programmiersprache.

In den folgenden Abschnitten werden die einzelnen Features aufgegriffen und deren Umsetzung vorgestellt. Dazu werden alle neuen Komponenten sowie die notwendigen Änderungen an Komponenten des ursprünglichen Concept Mappers beschrieben und auf Herausforderungen, die bei der Umsetzung aufkamen, eingegangen.

4.1. Speicherung von Concept Maps in der Moodle Datenbank

Im Folgenden werden generell, nach einmaliger Vorstellung, die Namen der Komponenten abgekürzt. Im *ConceptMapper1.0.js* wird in Intervallen die Concept Map gespeichert, außerdem wird wie gehabt beim Drücken von Speichern ebenfalls die aktuelle Concept Map als JSON formuliert und eine entsprechende Funktion des *StorageHandler.js* aufgerufen.

Im *StorageHandler* wird das Objekt, welches die Concept Map und alle Metadaten darstellt, als String umgewandelt und mit Hilfe von AJAX per asynchronem HTTP Request an die *lib.php* des Plug-Ins geschickt. Der content-type wurde für die Anfrage nicht gesetzt und somit wird auch die Concept Map auf der Seite des Servers als String behandelt. Hier werden ebenso zunächst die für die Speicherung notwendigen Metadaten aus dem Objekt gelesen und mitgeschickt. Dabei handelt es sich um die ID des Kursmoduls sowie die ID der Gruppe des Nutzers.

Dazu mussten im ConceptMapper entsprechend das Verhalten des Buttons zur Speicherung angepasst und die Intervallspeicherung ermöglicht werden. Der bestehende *StorageHandler*, welcher sich um das Speichern und Laden von Concept Maps kümmert, wurde um den *MoodleStorageHandler* erweitert. Um die oben beschriebene Funktion zu ermöglichen.

Nun wird in der *lib.php* zunächst die Anfrage interpretiert und die entsprechende Funktion zur Speicherung aufgerufen. In der Funktion *ccm_create_resource* wird zunächst ein Objekt *\$record* erstellt, welches als Anfrage-Record dient und später in den Moodle Funktionen zur Erstellung sowie Update von Einträgen in der Datenbank Verwendung findet. Hier befindet sich die vom *StorageHandler* erhaltene String-Repräsentation der Concept Map sowie Metadaten, die Kursmodul ID und die Gruppen ID.


```

metadata = {
  actor: //Benutzer
  {
    objectType,
    name, //neu | Moodle User-Name
    id, //neu | Moodle User-Id
    displayName,
    groupId, //neu
    canEdit, //neu
    role, //neu | teacher oder student
  },
  target: //Concept Map
  {
    objectType,
    id,
    dbId,
    displayName
  },
  generator: //Anwendung
  {
    objectType,
    url,
    id,
    displayName
  },
  provider: //Serverseitige Informationen
  {
    objectType,
    url,
    id, //neu | Moodle Course Module Id
    session, //neu | Schlüssel der TogetherJS Sitzung
    timeclose, //neu | Abgabedatum
  }
}

```

Abbildung 8 Überblick über die Metadaten

Anschließend wird auf eine existierende Concept Map in der Datenbank, bzw. genauer in der Tabelle *ccm_conceptmaps* geprüft. Existiert bereits ein Eintrag wird dieser lediglich aktualisiert. Existiert noch kein Eintrag wird ein neuer erstellt. Für beide Vorgehensweisen wird das eben erstellte Objekt *\$record* verwendet. Auf ähnliche Weise wurde das Laden von Concept Maps ermöglicht.

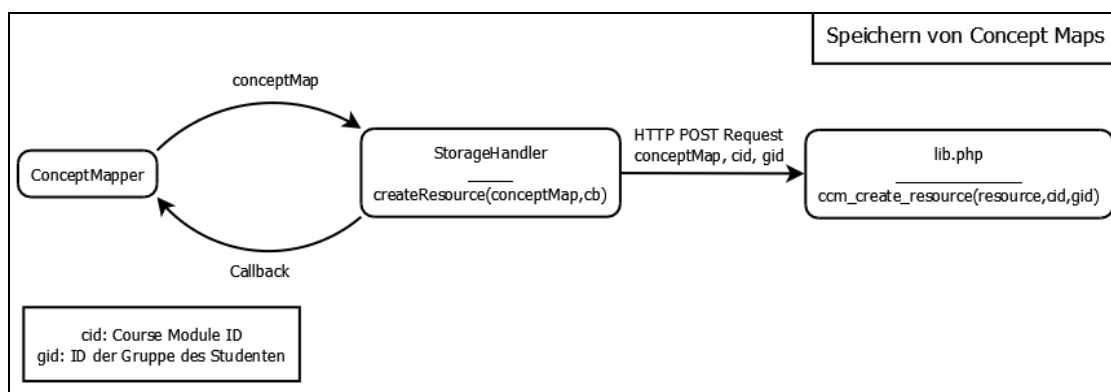


Abbildung 9 Speichern einer Concept Map

Zu sehen ist in dieser Abbildung der Ablauf bei der Speicherung von Concept Maps. Wie beschrieben verhält sich das Laden von Concept Maps sehr ähnlich.

In diesem Zusammenhang soll auch einmal der Aufbau der Tabellen in der Moodle Datenbank schematisch dargestellt werden. Die Spalte *id* als *sequence* ist ein Pflichtfeld in Moodle, welches eine fortlaufende ID für die einzelnen Einträge vergibt. *cid* und *gid* bezeichnen wie in Abbildung 9 schon zu sehen die Course Module ID sowie die ID der Gruppe des Studenten. Diese werden bei Anfragen verwendet um die richtige Concept Map aber auch das entsprechende Protokoll.

4. Implementierung

Das Protokoll ist in der Datenbank durch die Tabelle *ccm_actions* vertreten.

Sowohl die Concept Map als auch die einzelnen Aktionen liegen in der Datenbank im Binärformat vor und werden nach Ausgabe vom *StorageHandler* wieder als JSON interpretiert.

TABELLE	SPALTEN
ccm	{id, ccmid, course, name, timemodified, timedose}
ccm_conceptmaps	{id, gid, cid, skey map}
ccm_actions	{id, gid, cid, action}

Abbildung 10 Schematische Darstellung der Tabellen des Plug-Ins

4.2. Protokollierung sowie Ausgabe von Benutzeraktionen

Die Implementierung der Protokollierung erfolgt ähnlich wie das Speichern und Laden von Concept Maps. Benutzeraktionen werden von der Anwendung als *ActivityStream-Objekt* formuliert. Dieses enthält Informationen zum Typ der Benutzeraktion, Details über die Aktion, sowie Metadaten. Zu den Metadaten gehören unter anderem der Moodle Name des Nutzers, und seine Gruppen-ID.

Für das *loggen* der Aktionen ist der *ActionLogger* (*ActionLogger.js*) zuständig. Dieser wurde um die Funktion *togetherLogging()* erweitert, wo unter anderem die vom *ActionLogger* bereitgestellten *activityStreamObjects* an den *StorageHandler* weitergegeben werden. Abbildung 11 zeigt den Inhalt eines solchen *activityStreamObjects*. Auch wurde eine Funktion implementiert die ähnlich wie im vorherigen Absatz alle Informationen per AJAX an die *lib.php* weitergibt.

Zur Protokollierung gehört ebenfalls die Ausgabe der gespeicherten Benutzeraktionen. Dazu wurde eine Funktion in der *lib.php* implementiert, welche auf die entsprechende *ccm_actions* Tabelle zugreift, und alle Einträge von niedriger bis hoher ID ausliest. Nur die wichtigen Informationen wie der Nutzernamen, Typ und Inhalt der Aktivität werden anschließend ausgegeben. Die Anzeige der Protokolle erfolgt nach Prüfung der *Capabilities* in der *view.php*.

4.3. Verschiedene Sichten für die Nutzer des Plug-Ins

Dieser Abschnitt soll die Implementierung der unterschiedlichen Sichten des Plug-Ins erläutern. Die *view.php* ist der erste ausgeführte Teil eines Moodle Plug-Ins und in der Regel der Teil, welcher sich um die Darstellung des Inhaltes kümmert. Im Fall des hier entwickelten Plug-Ins wird in der *view.php* zunächst Informationen wie ID, Name, und Gruppen ID des Studenten gesammelt. Fallentscheidungen zur Darstellung werden wie im Ansatz beschrieben getroffen.

Diese gesammelten Daten und zusätzlich noch die Rolle des Teilnehmers und ob dieser editieren darf, werden an eine Funktion in der *lib.php* gesendet, welche letztendlich das *iframe* darstellt. Diese Informationen werden damit über die URL an den Concept Mapper übergeben, welche diese ausliest und in den Metadaten speichert. Zu beachten ist hierbei, dass TogetherJS, für eine volle Funktionalität, darauf besteht, dass alle Teilnehmer einer Sitzung dieselbe URL haben. Ist das nicht der Fall, weist TogetherJS darauf hin und überträgt weder Mausklicks, noch die Position des

Cursors. Deswegen werden im Query String der URL die *Gruppen-ID* sowie die *Course-Module-ID* übertragen und im Fragmentbezeichner kann der Moodle-Name des Teilnehmers, sowie seine ID, an den Concept Mapper übergeben werden.

Im *ConceptMapper* wird weiterhin, vor der Initialisierung der meisten Anwendungs-Komponenten geprüft, ob der Nutzer zum Editieren berechtigt ist. Ist das der Fall, so wird der Rest der Anwendung initialisiert. Darf der Nutzer die Concept Map nicht verändern, so wird nur die Concept Map aus der Datenbank geladen, aber die weiteren Funktionen und Event-Handler der Anwendung werden nicht initialisiert. Weiterhin werden per JQuery HTML Elemente auf nicht draggable gesetzt, um so eine reine statische Darstellung der erstellten Concept Map anzuzeigen. Für die Abgabe sowie Einsicht von Kommentaren ist eine Funktion in der *lib.php* zuständig. zusammen mit einer in *forms.php* definierten *Moodle-Form*. Die *selection.php* wird aufgerufen, wenn Lehrenden eine Auswahl zu allen erstellten Concept Maps einer Aktivität gegeben werden soll. Hier werden durch eine *form* die IDs der Gruppen in einer drop-down Auswahl angezeigt.

4.4. Initialisierung der TogetherJS Sitzung

Die Initialisierung der TogetherJS Sitzung wird im *ConceptMapper* umgesetzt. Dazu wird zunächst über *StorageHandler* und *lib.php* geprüft, ob bereits eine Concept Map in der Tabelle vorhanden ist. Die Prüfung erfolgt über Kursmodul- sowie Gruppen-ID (*cid*, bzw. *gid* respektive). Ist solch ein Eintrag vorhanden wird die entsprechende Concept Map sowie die dazugehörigen Metadaten geladen. Dazu mussten für die Funktionen *resourceExists()* des *StorageHandlers* sowie *loadConceptMap()* des *ConceptMappers* entsprechende *Callbacks* realisiert werden.

In den Metadaten ist unter anderem auch der Sitzungsschlüssel gespeichert, aus dem nun, mittels *TogetherJS.config("findRoom",key)* und dem Start von TogetherJS, eine entsprechende Sitzung mit diesem Schlüssel eingerichtet oder beigetreten wird.

Ist noch kein Eintrag vorhanden, hat auch noch kein Teilnehmer in einer Gruppe in dieser Instanz des Plug-Ins eine Concept Map bearbeitet. Es wird in diesem Fall ein zufälliger Schlüssel generiert und in den Metadaten gespeichert. Anschließend wird der Eintrag der Daten in die Tabelle vorgenommen.

Aufbau: Nachricht	TogetherJS.send	activityStreamObject =	Aufbau: activityStream Objekt
	<pre> ({ type, conceptId, label, colorName, className, xPos, yPos }) </pre>	<pre> { published, //aktuelles Datum actor, //Benutzer verb, //Typ der Aktion object, target, generator, provider } </pre>	

Abbildung 11 Nachrichtenaufbau und Aufbau des activityStreamObjects

4.5. Synchronisierung von Benutzeraktionen

Hier und in den folgenden Abschnitten wird die Implementierung der Synchronisierung der Darstellung und damit die Ermöglichung der kollaborativen Zusammenarbeit erläutert. Das Prinzip des Nachrichtenaustauschs mit Hilfe von TogetherJS wurde bereits im Ansatz dargestellt, im Folgenden soll die Implementierung dessen beschrieben werden.

Im *ConceptMapper* werden durch Benutzeraktionen ausgelöste Events entsprechend behandelt. Die Aktion wird durchgeführt und es wird ein Objekt erstellt welches Informationen über die Aktion des Nutzers enthält. Hier mussten diese Objekte um weitere Details erweitert werden, damit die Aktionen nur aus diesen Informationen wiederholt werden können. So musste z.B. das sogenannte Verb *concept* um x-, y-Position des Konzepts sowie *colorName* und *className* erweitert werden.

Entsprechende Aktionen werden über *_logAction()* an den *ActionLogger* übergeben, hier wird in der bereits erwähnten neuen Funktion *togetherLogging()* der Nachrichtenversand durchgeführt. Dafür bietet TogetherJS mit der Funktion *TogetherJS.send()* eine einfache Möglichkeit des Nachrichtenversands an allen Teilnehmer in der schon bestehenden Sitzung. Zu sehen ist in Abbildung 11 der Aufbau einer über TogetherJS übermittelten Nachricht. Zunächst wird hier überprüft ob TogetherJS läuft bevor im Anschluss eine Nachricht mit einem bestimmten Typ und weiteren Attributen versandt wird.

Nun wird im *TogetherHandler.js* die erhaltene Nachricht abgefangen. Dabei wird durch *TogetherJS.hub.on()* das ausgelöste Event entsprechend nach dem *type* welcher im *ActionLogger* angegeben wurde behandelt. Dies ist für alle Aktionen so implementiert, dass diese mit gewissen Anpassungen wiederholt werden.

So wird z.B. beim *type addConcept* einfach nur die angepasste *_createConcept()* Funktion des *ConceptMappers* mit den erhaltenen Informationen aufgerufen. Bei *updateConcept* muss jedoch das bestehende Konzept gefunden und entsprechend die neue Position gesetzt werden. Auch muss die Funktion *_createConcept()* noch weiter angepasst werden. Es sollen nur Konzepte erstellt werden, deren ID nicht bereits in der Concept Map existieren, dazu wurde eine Funktion implementiert welche dies prüft. Würde die Prüfung nicht erfolgen, so würden Duplikate erstellt, da bei der Wiederholung einer Aktion eines anderen Clients ebenfalls wieder die Aktion 'geloggt' wird und einen erneuten Nachrichtenversand anstößt.

4.6. Behandlung von neuen Teilnehmern in einer Sitzung

Treten neue Teilnehmer einer Sitzung bei, soll, wie im Ansatz herausgearbeitet, von anderen Teilnehmern die aktuelle Concept Map geschickt werden. Die vom neuen Client über TogetherJS gesendete *hello*-Nachricht wird bei den anderen Sitzungsteilnehmern empfangen und es wird die aktuelle Concept Map als JavaScript Objekt gesammelt und mit dem *type loadMap* versandt. Bevor das geschieht wird zunächst jedoch überprüft, ob beim Client gerade eine Map geladen wird oder wurde, damit keine leere Map übertragen wird. Umgesetzt wird es durch Überprüfung der Variablen *readyForMessages* und *init*, welche angeben ob gerade eine Concept Map durch *setConceptMapFromJSON()* geladen wurde oder die Anwendung noch nicht lange initialisiert ist. Erhält der neue Client eine Nachricht mit dem *loadMap* wird diese entsprechend geladen.

4.7. Umgang mit Übertragungsfehlern und konkurrierenden Aktionen

Das Plug-In soll wie im Ansatz beschrieben mit Übertragungsfehlern bzw. allgemein mit unterschiedlichen Darstellungen bei den einzelnen Teilnehmer umgehen können. Dazu wird in Intervallen innerhalb des ConceptMappers die Concept Map über TogetherJS verschickt, sofern, wie schon im letzten Abschnitt, *init* sowie *readyForMessages* wahr sind. Auf diese Weise wird keine leere Map gesendet. Die Nachricht löst wiederum bei anderen Teilnehmern ein hub Event mit dem Typ 'loadMap' aus.

Hier wird zunächst über die Variable *readyForMessages* überprüft, ob nicht bereits schon eine Concept Map geladen wird zu dieser Zeit. Ist *readyForMessages* wahr, wird die Variable zunächst auf *false* gesetzt und im Anschluss werden die Strings beider JavaScript Objekte, welche die Concept Map vertreten, auf ihre Gleichheit überprüft. Dies wird gemacht, damit nicht unnötigerweise die Concept Map geladen wird, sondern nur bei tatsächlichen Unstimmigkeiten in den Maps. Natürlich kann sich bei den einzelnen Clients die Reihenfolge der erstellten Konzepte, Verbindungen, etc. in der Concept Map voneinander unterscheiden. Solch ein *false-positive* hat aber nur den Update der Concept Map zur Folge und keine größeren negativen Konsequenzen. Sind jedenfalls die eigene sowie die erhaltene Concept Map unterschiedlich, so wird die Funktion *setConceptMapFromJSON()* aufgerufen und so die Concept Map geladen.

Diese bestehende Funktion musste jedoch erweitert werden, um den reibungslosen Ablauf beim Update der Map zu gewährleisten. Es wird zunächst wie gehabt die aktuelle Concept Map gelöscht, jedoch wird das Laden der neuen Map verzögert. Der Grund dafür ist, dass bei dem Erstellen von Konzepten ohne diese Verzögerung die alte Concept Map noch nicht gelöscht ist. Der erste Ansatz dazu, die Verwendung von Callbacks, hatte nicht das gewünschte Ergebnis liefern können. Zusätzlich wird nun in dieser Funktion auch überprüft, ob Verbindungen bereits existieren.

Wird die Verzögerung zwischen dem Löschen der Map und dem Wiederherstellen nicht vorgenommen, so kann es, mit der Überprüfung auf existierende Konzepte und Verbindungen, dazu kommen dass die Concept Map gelöscht und nicht wieder geladen wird. Ohne die Überprüfung können an dieser Stelle Duplikate von Konzepten und Verbindungen entstehen.

Ändern bei dieser Lösung Studenten den Text eines Konzepts oder einer Verbindung, während des Sendens der eigenen Concept Map, so kann es sein, dass ein leerer String übertragen wird. Damit erhalten andere Teilnehmer eben diesen leeren String, während man selbst entsprechend den eingegebenen Text darstellt. Solch eine Inkonsistenz wird jedoch spätestens beim der nächsten Synchronisation behoben.

Wie, und wie gut nun diese Implementierung in der Praxis funktioniert, soll im nächsten Kapitel überprüft werden.

4. Implementierung

5. Evaluation

In diesem Kapitel soll nun, nachdem alle Funktionen und deren Implementierung besprochen wurden, das Ergebnis vorgestellt werden.

Dazu wird Bezug auf die im Ansatz beschriebenen Anwendungsfälle genommen. Es wird eine Aufgabenstellung formuliert und die Bearbeitung dieser Aufgabe mit der Unterstützung von Bild-Ausschnitten aus dem Plug-In gezeigt. Dies soll den Zweck haben, das Plug-In vorzustellen und kritisch auf seine Funktionalität hin zu untersuchen. Abschließend wird TogetherJS und die dadurch gebotenen Funktionen für die Verwendung im Plug-In hinsichtlich der Funktionalität betrachtet.

5.1. Anwendungsfälle

Es soll hier eine mögliche Sitzung zur kollaborativen Bearbeitung einer Aufgabe vorgestellt werden. Dazu wird zunächst eine Aufgabe formuliert.

Das Eberly Center der Carnegie Mellon Universität bietet in seiner Webpräsenz Hinweise zur Benutzung von Concept Maps in der Lehre an⁶. Die folgende Aufgabenstellung wurde mit Hilfe dieser Hinweise erstellt.

Aufgabe:

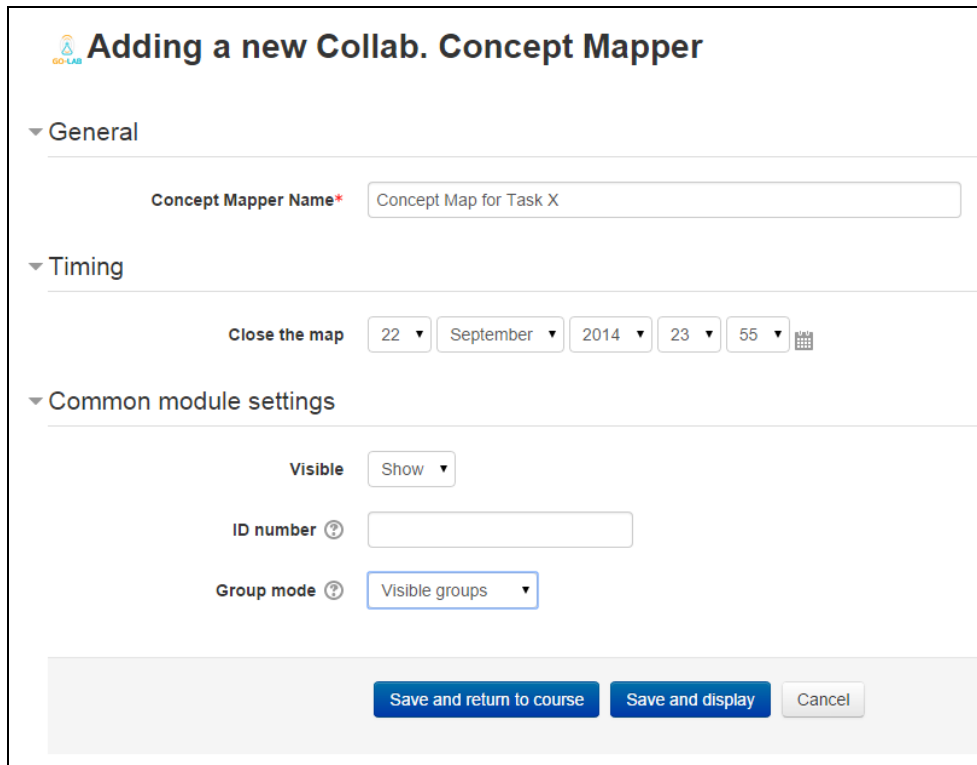
Erläutern Sie das Säure-Base-Prinzip mit Hilfe einer Concept Map. Dabei sollen wichtige Reaktionen, Begriffe, sowie Verbindungen dazwischen dargestellt werden. Überlegen Sie sich dazu zunächst, am besten zusammen in der Gruppe, alle Konzepte die Ihnen dazu einfallen. So haben Sie eine gemeinsame Basis um die Concept Map weiter zu entwickeln. Nehmen Sie sich die Zeit die Concept Map in mehreren Sitzungen zu bearbeiten.

Im Folgenden werden die vier bereits im Ansatz beschriebenen Anwendungsfälle im Zusammenhang mit dem umgesetzten Plug-In vorgestellt. Es werden die implementierten Funktionen kurz vorgestellt und besonders auf mögliche Schwachstellen in der Implementierung bzw. mögliche Probleme im Umgang mit dem Plug-In eingegangen.

⁶ <https://www.cmu.edu/teaching/assessment/assesslearning/conceptmaps.html>

5. Evaluation

5.1.1. Lehrende - Innerhalb des Bearbeitungszeitraums



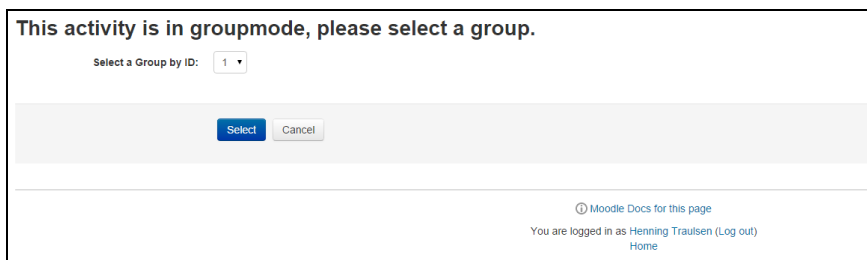
The screenshot shows a Moodle form titled "Adding a new Collab. Concept Mapper". It is divided into three sections: "General", "Timing", and "Common module settings".

- General:** Contains a text field for "Concept Mapper Name*" with the value "Concept Map for Task X".
- Timing:** Contains a "Close the map" section with date and time pickers: "22", "September", "2014", "23", and "55".
- Common module settings:** Contains a "Visible" section with a "Show" dropdown, an "ID number" field with a help icon, and a "Group mode" dropdown set to "Visible groups".

At the bottom, there are three buttons: "Save and return to course", "Save and display", and "Cancel".

Abbildung 12 Konfiguration des Plug-Ins

Zu sehen sind in Abbildung 12 die Konfigurationsmöglichkeiten, die das Programm bietet. Über Timing kann das Abgabedatum festgelegt werden und unter den allgemeinen Einstellungen kann der Gruppenmodus eingestellt werden.



The screenshot shows a modal dialog titled "This activity is in groupmode, please select a group." It contains a "Select a Group by ID:" label and a dropdown menu with the value "1". Below the dropdown are "Select" and "Cancel" buttons. At the bottom right, there is a link to "Moodle Docs for this page" and a login status message: "You are logged in as Henning Traulsen (Log out) Home".

Abbildung 13 Auswahl der Gruppe

Dem Lehrenden werden im Gruppenmodus zunächst die bisher erstellten Concept Maps angezeigt. Die Auswahl erfolgt über die ID der jeweiligen Gruppe.

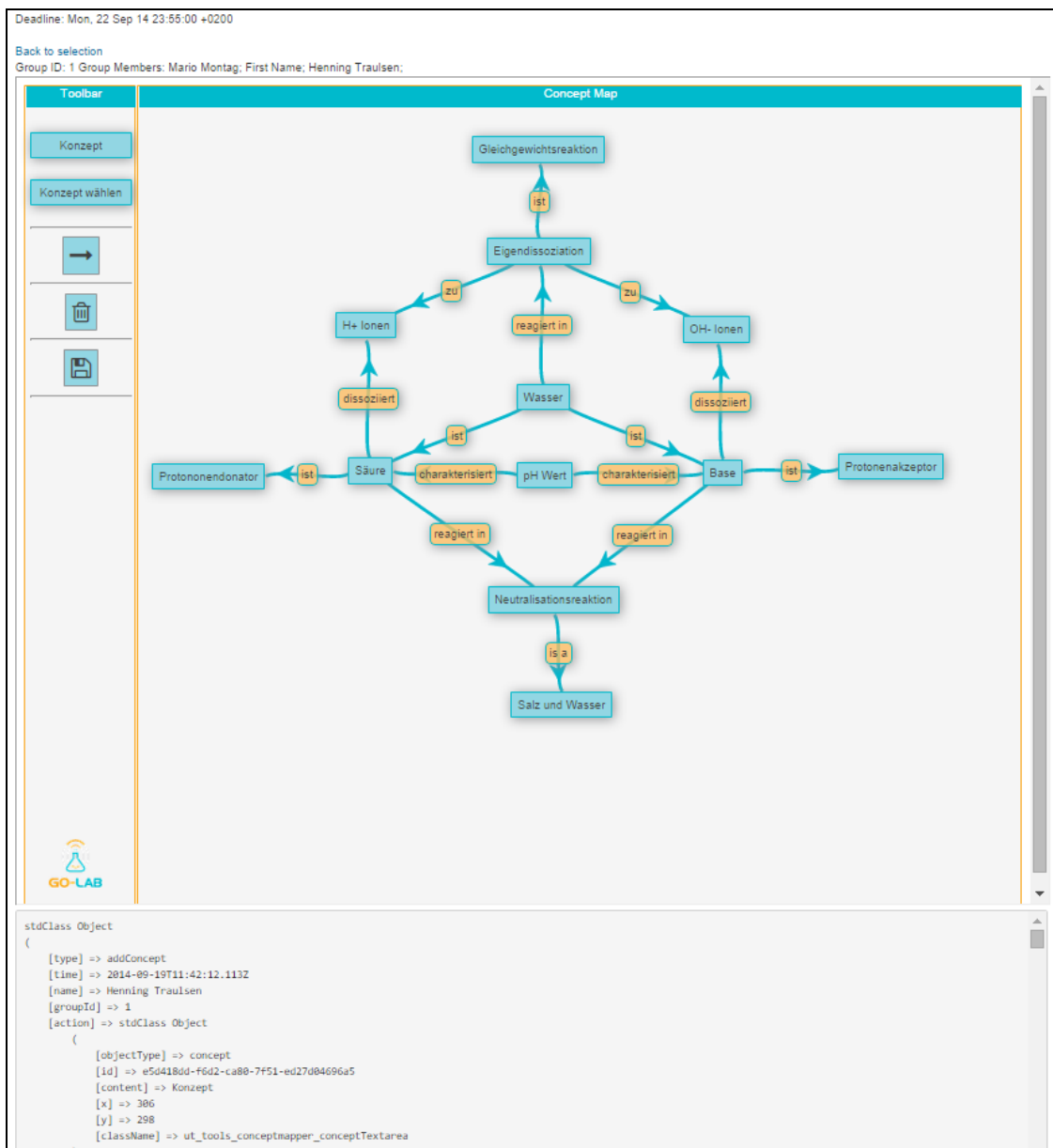


Abbildung 14 Lehrende - Innerhalb des Bearbeitungszeitraums

Lehrende können hier nach Auswahl der Map die Concept Map ansehen, jedoch nicht bearbeiten. Das Protokoll der Benutzeraktionen ist einsehbar.

5. Evaluation

5.1.2. Studenten - Innerhalb des Bearbeitungszeitraums

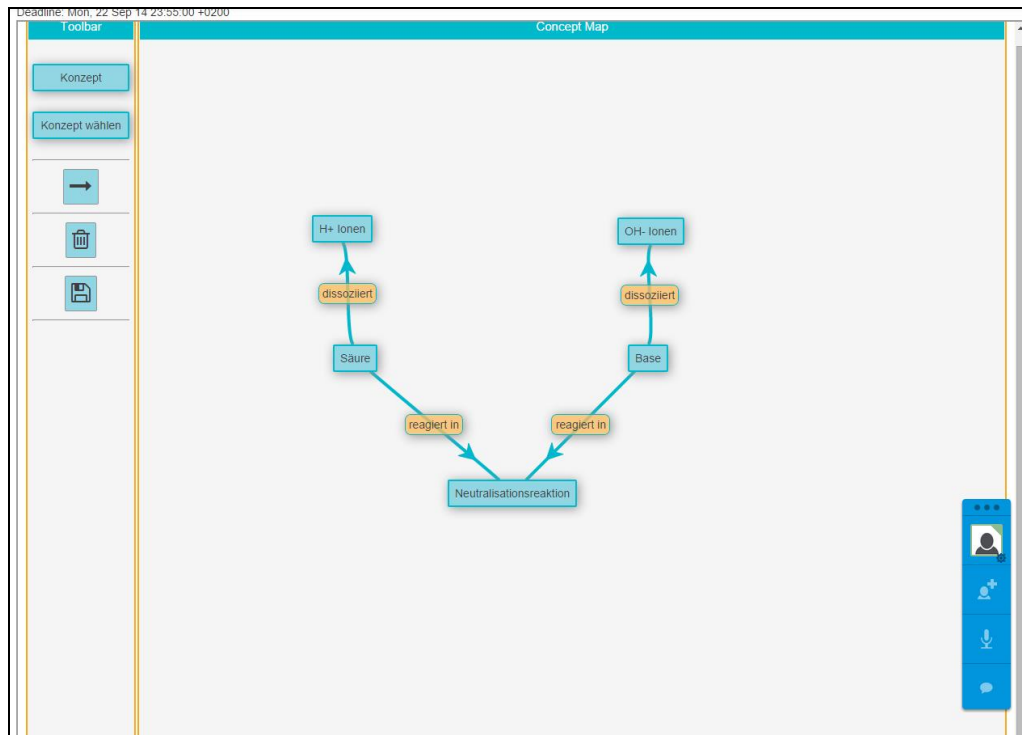


Abbildung 15 Ein Student bearbeitet die Concept Map

Der Student bearbeitet zunächst in Einzelarbeit eine bestehende Concept Map. Hier wurde in einer früheren Sitzung bereits eine Concept Map bearbeitet und gespeichert. Der Student ist an dieser Stelle der richtigen Sitzung entsprechend seiner Concept Map beigetreten und hat auch die entsprechende Concept Map geladen.

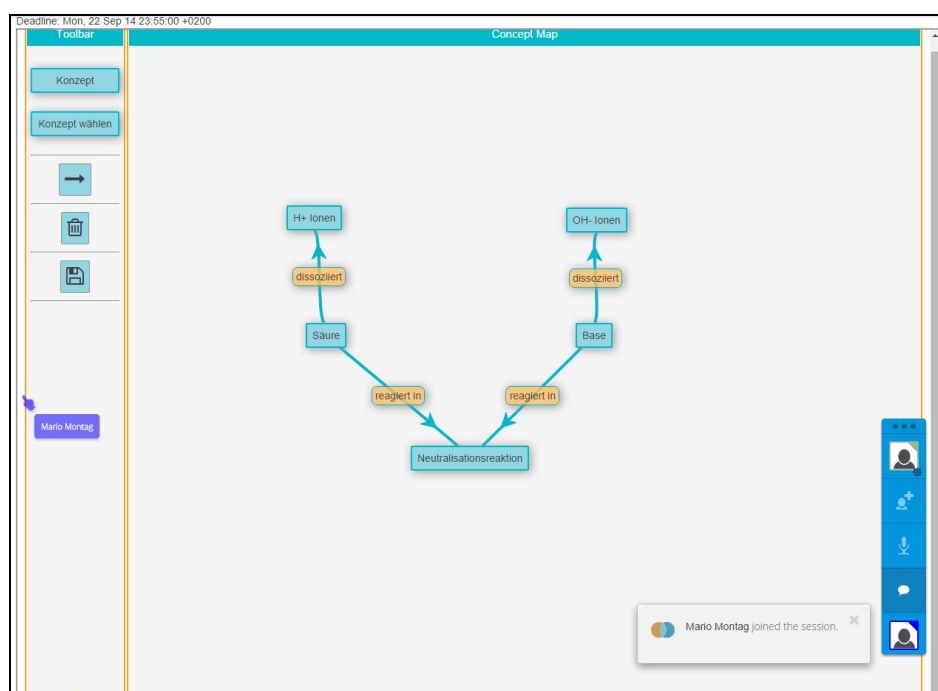


Abbildung 16 Zweiter Student tritt der Sitzung bei

Kommt nun in Abbildung 16 ein zweiter Teilnehmer zur kollaborativen Sitzung hinzu wird entsprechend der erste Teilnehmer seine Concept Map an ihn senden. Anschließend können beide gemeinsam die Concept Map bearbeiten.

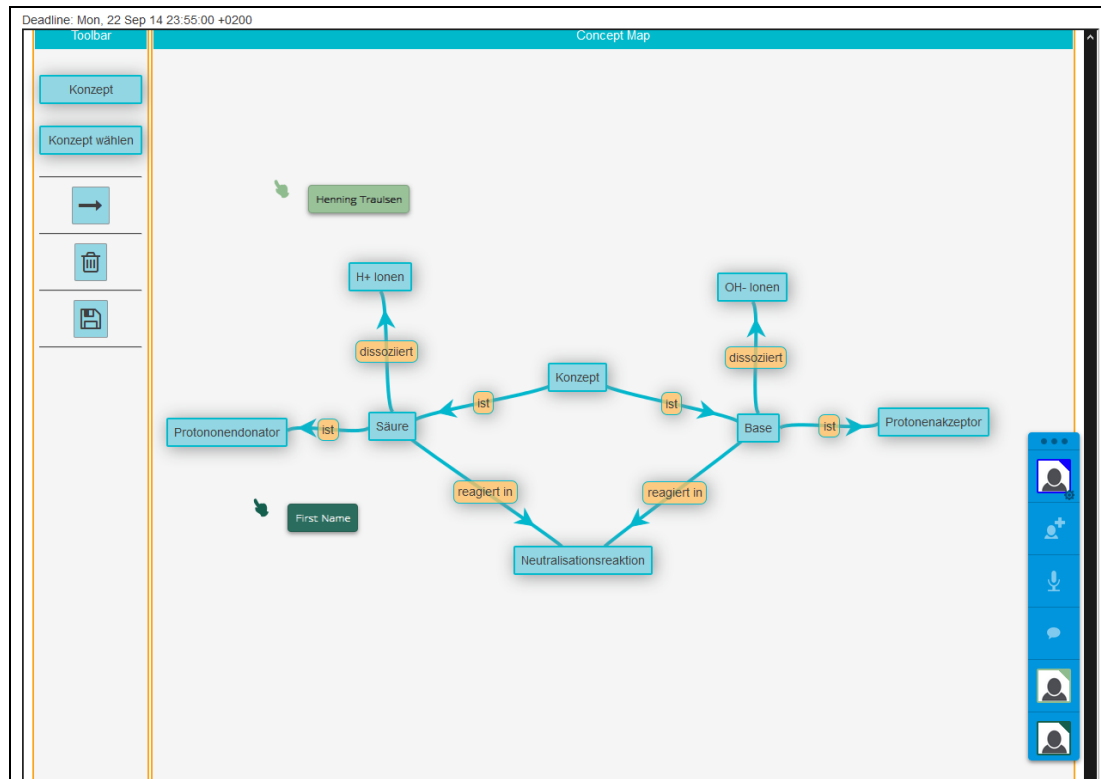


Abbildung 17 Drei Studenten bearbeiten die Concept Map

Käme jetzt, anders als hier dargestellt, kurz danach ein Dritter hinzu, wird der zweite seine Map noch nicht senden um zu verhindern, dass eine leere Concept Map gesendet wird. Der erste Teilnehmer sorgt hier wieder dafür, dass die Darstellung synchron ist.

Hier kann natürlich auch der Fall vorkommen, dass der erste Teilnehmer einer Sitzung die Map bereits bearbeitet hat und kurz danach ein zweiter Teilnehmer hinzukommt. Da Clients ihre Map nur versenden, wenn sie nicht gerade selbst eine Map geladen haben bzw. gerade erst initialisiert wurden, kann es dazu kommen, dass der neue Teilnehmer keine Antwort bekommt. In diesem Fall sind die dargestellten Concept Maps unterschiedlich und es kommt erst nach der entsprechenden Intervallzeit (2 Minuten) zur Synchronisation der Concept Maps.

Der Austausch von Benutzeraktionen unter allen Teilnehmern scheint, bis auf entsprechende Fehler bei der Übertragung, reibungslos zu funktionieren. Das implementierte Prinzip zur Synchronisation der kompletten Concept Map sowie bei der Initialisierung hat jedoch noch eventuelle Schwachstellen.

So kann es dazu kommen, dass Maps geladen werden, obwohl sie identisch mit der aktuellen sind. Das führt zu einer unnötigen Störung, die sich jedoch nur über wenige Wiederholungen der Synchronisierung hinweg ziehen.

Die notwendige Verzögerung zwischen dem Löschen der aktuellen, und dem Laden der neuen Concept Map führt zu einem störenden *blinken* der gesamten Map, so sollten diese unnötigen Wiederherstellungsvorgänge vermieden werden.

5. Evaluation

Die Variable *readyForMessages* sorgt dafür, dass immer nur eine Concept Map wiederhergestellt wird. Gerade bei der Initialisierung neuer Teilnehmer ist das wichtig, um nicht nur das eben angesprochene *blinken* der Map zu vermeiden, sondern vielmehr damit verhindert wird, dass die Concept Map nach dem Löschen nicht wiederhergestellt wird.

Das Prinzip, dass letztendlich immer nur eine Map geladen wird, könnte weiterentwickelt werden. So wäre ein System, welches immer einen Führer in der Sitzung hält eine solche Weiterentwicklung. Dazu könnten theoretisch die Variablen *isClient* bzw. *isCreator* die jedem Client durch TogetherJS zugeordnet wird genutzt werden. Jedoch bietet TogetherJS die nötige Verwaltung dafür nicht an, diese müsste noch für jeden Client implementiert werden. Solch ein System würde zu weniger genutzter Bandbreite führen und auch lokal beim Client Ressourcen sparen.

Das sind die Kritikpunkte, die nach einigen Tests am Plug-In anzumerken sind. Insgesamt ist aber zu erwarten, dass die kollaborative Bearbeitung von Concept Maps mit der aktuellen Implementierung ohne Probleme durchführbar ist.

5.1.3. Lehrende - Außerhalb des Bearbeitungszeitraums

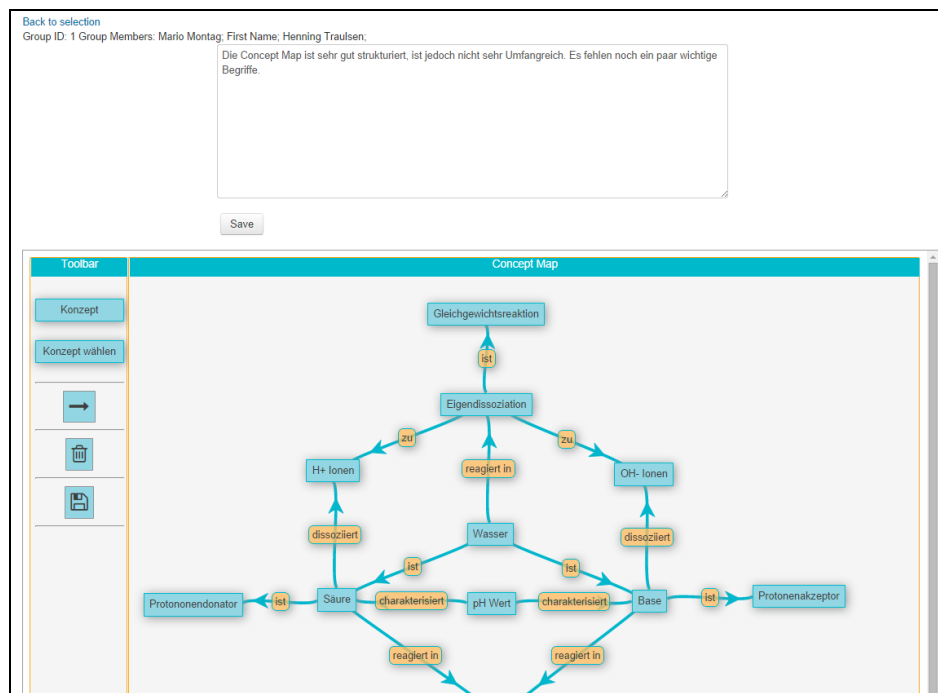


Abbildung 18 Lehrend - Außerhalb des Bearbeitungszeitraums

Die für den Lehrenden angezeigte Darstellung ist, bis auf die neue Kommentarfunktion, dieselbe wie die Darstellung innerhalb des Bearbeitungszeitraums. Hier dargestellt ist nur ein Ausschnitt der einzusehenden Concept Map sowie die Kommentarfunktion. Es kann nun über dieses Kommentarfeld ein Feedback zu der erbrachten Leistung gegeben werden. Der Kommentar kann von Lehrenden beliebig oft geändert werden.

5.1.4. Studenten - Außerhalb des Bearbeitungszeitraums

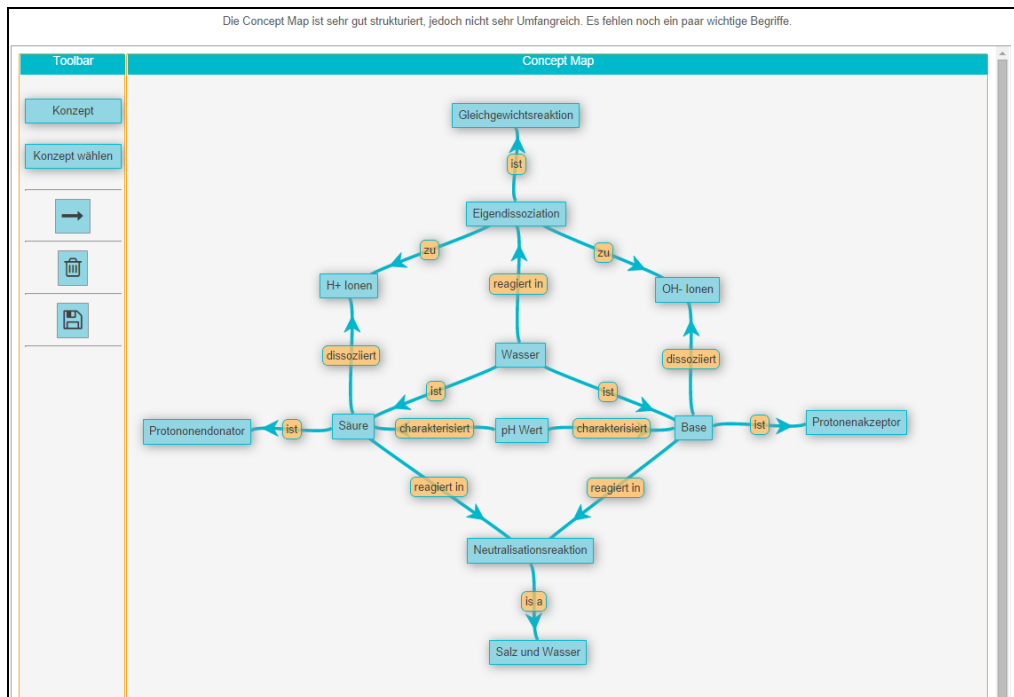


Abbildung 19 Studenten - Außerhalb des Bearbeitungszeitraums

Außerhalb des Bearbeitungszeitraums ist die Concept Map auch für den Studenten nicht editierbar. Der durch den Lehrenden abgegebene Kommentar kann nun oberhalb der Concept Map eingesehen werden.

5.2. Mögliche Probleme mit TogetherJS

TogetherJS behält Informationen über eine Sitzung innerhalb eines Tabs. Selbst wenn die Aktivität verlassen und die Seite neu geladen wird. Wird also nach dem Verlassen einer Concept-Mapper-Aktivität im selben Tab eine andere solche Aktivität geöffnet, so baut TogetherJS unabhängig von dem übergebenen Sitzungsschlüssel die Verbindung zur vorherigen Sitzung wieder auf. Da Lehrpersonen keiner Sitzung beitreten stellt dieser Umstand in diesem Fall kein Problem dar. Es könnten jedoch damit Probleme auftreten, wenn Studenten im selben Tab des Browsers eine andere Concept-Mapper-Aktivität aufrufen.

5. Evaluation

6. Zusammenfassung und Ausblick

6.1. Zusammenfassung

In dieser Arbeit wurde erfolgreich ein Moodle Plug-In zur kollaborativen Bearbeitung von Concept Maps erstellt. Es wurden Grundlagen vorgestellt, daraus Anforderungen herausgearbeitet und die notwendigen Funktionen implementiert. Das in dieser Arbeit entwickelte Plug-In bietet alle gewünschten Konfigurierungsmöglichkeiten und kann wie jedes Modul in Moodle leicht installiert und instanziiert werden. Getrennte Sichten für Lehrende und Lernende ermöglichen die Trennung der Funktionalitäten.

Bei der Instanziierung des Plug-Ins ist es Lehrenden möglich ein Abgabetermin festzulegen und die Unterstützung von Moodle Gruppen auszuwählen. Damit können Studierende während des Bearbeitungszeitraums gemeinsam innerhalb von Gruppen eine Concept Map bearbeiten. Jede Gruppe bearbeitet so für jede Instanz des Moduls ihre eigene Map. Ist der Abgabetermin erreicht bzw. überschritten, so ist es Lernenden nicht mehr möglich die Concept Map zu bearbeiten. Lehrende können zu diesem Zeitpunkt durch einen Kommentar ihr Feedback zur abgegebenen Concept Map geben. Die Protokollierung von Benutzeraktionen wird in der Moodle Datenbank durchgeführt und es wurde eine Ansicht dieses Protokolls für Lehrende ermöglicht.

Studierenden wird es durch das Plug-In ermöglicht Concept Maps zu erstellen. Insbesondere ist es Studierenden nun möglich diese Concept Maps in einer kollaborativen Sitzung gemeinsam zu bearbeiten. Dafür wurde die Speicherung der Concept Maps in der Datenbank ermöglicht um so auch die asynchrone Bearbeitung zu unterstützen. Studierende können so die Concept Map speichern und zu einem späteren Zeitpunkt wieder bearbeiten, solange der Abgabetermin noch nicht erreicht ist. Die Zwischenspeicherung von Concept Maps sorgt dafür, dass in gewissen Fällen nur wenige Daten verloren gehen.

Durch Austausch von Aktionen zur Erstellung der Concept Map, unter allen Teilnehmer einer Sitzung, wird die kollaborative Bearbeitung von Concept Maps durch Studierende ermöglicht. Ebenso werden neue Teilnehmer einer Sitzung auf den aktuellen Stand der Bearbeitung gebracht und es wurden Maßnahmen vorgenommen, um Synchronisierungsfehler zu beheben.

6.2. Ausblick

Verbesserungen der Implementierung, Weiterentwicklung der Funktionen, sowie neue Features sind für die weitere Entwicklung des Plug-Ins vorstellbar. Die im Folgenden vorgestellten Features konnten aus Zeitgründen nicht mehr mit in die Arbeit eingebracht werden.

So wäre das schon angesprochene System der Sitzungsverwaltung, welches einem Teilnehmer die Rolle des Leiters zuordnet, eine wichtige Verbesserung bei der Initialisierung von neuen Teilnehmern und der Behandlung von Kommunikationsfehlern. Es werden weiter noch teilweise unnötigerweise Concept Maps geladen, was zu der schon beschriebenen Störung führen kann. Hier wäre ein besserer Vergleich der Concept Maps ein Ansatz, um dies zu verbessern.

6. Zusammenfassung und Ausblick

Neben dem Speichern der Maps in Intervallen würde das Speichern beim Schließen des Tools bzw. des Tabs weiter vor Datenverlust schützen.

Im Rahmen von Moodle könnte man sich auch eine bessere Kommentarfunktion wünschen, oder die Funktion von Moodle zur Abgabe von Bewertungen nutzen möchten.

Weitere Konfigurationsmöglichkeiten neben Abgabetermin und Gruppenmodus sind vorstellbar. Wie zum Beispiel die Einstellung, welche Nutzergruppe wann die Concept Map einsehen und editieren darf. Dazu müssten auch Änderungen im Concept Mapper vorgenommen werden.

Ebenso wäre die Überprüfung auf den Bearbeitungszeitraum innerhalb des Concept Mappers, und nicht nur bei dessen Aufruf, eine leichte jedoch wichtige Verbesserung.

Der Wegfall der Auswahlseite für Lehrende wäre ebenfalls denkbar. So kann die Auswahl auch in derselben Ansicht wie die Darstellung der Concept Map erfolgen. Dabei würde per *default* die erste erstellte Map geladen werden.

Einige Verbesserungen hinsichtlich der gemeinsamen Bearbeitung von Concept Maps können in Form von der Hervorhebung von ausgewählten Konzepten und der Anzeige, welches Label gerade bearbeitet wird, erfolgen. Weiter wäre es gut, die Einlade-Funktion von TogetherJS zu deaktivieren, sowie die Möglichkeit zur Beendigung von Sitzungen in der TogetherJS-Leiste zu unterbinden.

Neben diesen vielen kleineren Verbesserungen und Weiterentwicklungen des Plug-Ins gibt es auch noch größere Herausforderungen, welchen man sich während der weiteren Entwicklung zuwenden kann. Die Möglichkeit der Ausgabe sowie Eingabe von Concept Maps in gängigen Dateiformaten wäre so eine Herausforderung. Auch die Ausgabe bzw. Auswertung der Protokolle von Benutzeraktionen fällt in diese Kategorie.

Weiter wäre eine andere sehr interessante Funktionalität die Möglichkeit die erstellten Concept Maps durch eine *Zeitleiste*, oder etwas vergleichbares, zu rekonstruieren.

Literaturverzeichnis

Balter, R., Atallah, S. B. & Kanawati, R., 1995. Architecture for Synchronous Groupware Application Development.

Bascones, J. & Novak, J., 1985. *Alternative instructional systems and the development of problem-solving skills in physics*, s.l.: European Journal of Science Education.

Bonk, C. & Zhang, K., 2006. Introducing the R2D2 model: Online learning for the diverse learners of this world. Distance Educatio.

De Simone, C., Schmid, R. & McEwen, L., 2001. Supporting the learning process with collaborative concept mapping using computer-based communication tools and processes.. *Educational Research and Evaluation*.

Derry, S., Gance, S., Gance, L. L. & Schlager, M., 2000. Toward assessment of knowledge-building practices in technology-mediated work group interaction.

Webpräsenz des Eberly Center der Carnegie Mellon University. [Online]
Available at: <https://www.cmu.edu/teaching/assessment/assesslearning/conceptmaps.html>
[Zugriff 2014].

Eppler, M. J., 2006. A comparison between concept maps, mind maps, conceptual diagrams, and visual metaphors as complementary tools for knowledge construction and sharing. *Information Visualization*.

Eseryel, D., Ganesan, R. & Edmonds, G. S., 2002. Review of Computer-Supported Collaborative Work Systems. *Educational Technology & Society*.

Gao, H., Shen, E., Losh, S. & Turner, J., 2007. A Review of Studies on Collaborative Concept Mapping: What Have We Learned About the Technique and What Is Next? *Jl. of Interactive Learning Research*

Garrison, D. & Kanuka, H., 2004. Blended learning: Uncovering its transformative potential in higher education. *The Internet and Higher Education*.

Go-Lab-Project, Project, Zugriff 18 September 2014
Available at: <http://www.go-lab-project.eu/project>

Grudin, J., 1994. CSCW: History and Focus.

Gunawardena, C., Lower, C. & Anderson, T., 1997. Analysis of global online debate and the development of an interaction analysis model for examining social construction of knowledge in computer conferencing.

- Gurlitt, J. & Nückles, M., 2010. Kann man "Lernen Lernen" lehren? Erkenntnisse der Instruktionsforschung über Lernstrategien..
- Gutwin, C. & Greenberg, S., 2002. A descriptive framework of workspace awareness for real-time groupware. *Computer Supported Cooperative Work*.
- Higley, M., 2013. Benefits of Synchronous and Asynchronous e-Learning.
- Johnson, D. W., Johnson, R. T. & Smith, K. A., 1991. Cooperative learning: increasing college faculty instructional productivity.
- Johnson, R. T., Johnson, D. W. & Holubec, E. J., 1994. Cooperative Learning in the classroom.
- Kozar, O., 2010. Towards better group work: Seeing the difference between cooperation and collaboration. *English Teaching Forum*.
- Lipponen, L., 2002. Exploring foundations for computer-supported collaborative learning.
- Lynch, K. J., Snyder, J. M., Vogel, D. R. & McHenry, W. K., 1990. Supporting Collaborative Research on International Technological Trends.
- Ostwald, J., 1996. *Knowledge construction in software development: The evolving artifact approach*.
- Suthers, D. D., 2001. Architectures for Computer Supported Collaborative Learning.
- Van Zele, E., Lenaerts, J. & Wieme, W., 2004. Improving the Usefulness of Concept Maps as a Research Tool for Science Education. *International Journal of Science Education*, pp. 1043-1064.
- Xie, Y. & Sharma, P., 2011. Examining students reflective thinking from keywords tagged to blogs: using map analysis as a content analysis method. *Interactive Learning Environments*.

A. Eidesstattliche Erklärung zur Bachelorarbeit

Name: Traulsen

Vorname: Henning

Ich versichere an Eides statt durch meine untenstehende Unterschrift,

- dass ich die vorliegende Arbeit - mit Ausnahme der Anleitung durch die Betreuer - selbstständig ohne fremde Hilfe angefertigt habe und
- dass ich alle Stellen, die wörtlich oder annähernd wörtlich aus fremden Quellen entnommen sind, entsprechend als Zitate gekennzeichnet habe und
- dass ich ausschließlich die angegebenen Quellen (Literatur, Internetseiten, sonstige Hilfsmittel) verwendet habe und
- dass ich alle entsprechenden Angaben nach bestem Wissen und Gewissen vorgenommen habe, dass sie der Wahrheit entsprechen und dass ich nichts verschwiegen habe.

Mir ist bekannt, dass eine falsche Versicherung an Eides Statt nach § 156 und nach § 163 Abs. 1 des Strafgesetzbuches mit Freiheitsstrafe oder Geldstrafe bestraft wird.

Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Ort, Datum

Unterschrift