

Pflichtenheft

Mark Bigler, Lukas Zbinden und Simon Raess

17. März 2005

Inhaltsverzeichnis

1	Einleitung	4
1.1	Zweck des Dokumentes	4
2	Ausgangslage	4
3	Ist-Zustand	4
4	Ziele	4
4.1	Aufbau von Knowhow im Bereich CSCW	5
4.2	Bestehende Algorithmen evaluieren	5
4.3	Algorithmus implementieren	5
4.4	Testframework für Algorithmus	5
4.5	Konzept GUI	5
4.6	Konzept Netzwerk/Kommunikation	6
4.7	User Stories für kollaborativen Texteditor	6
5	Anforderungen	6
5.1	Architektur	6
5.2	Schnittstellen	6

Versionskontrolle

Version	Datum	Verantwortlich	Bemerkungen
0.1	15.03.2005	zbinl	Erste Version
0.2	16.03.2005	Projektteam	Überarbeitung

Tabelle 1: Versionskontrolle

	Stelle	Datum	Visum	Bemerkungen
Freigegeben	Projektteam			
Genehmigt				

Tabelle 2: Prüfung/Genehmigung

1 Einleitung

Das Projekt ACE entwickelt einen kollaborativen, plattformunabhängigen Editor. Diese Applikation soll mehreren Personen ermöglichen, ein Textdokument gemeinsam zu bearbeiten. Dabei arbeitet jede Person mit dem Editor an einem eigenen Computer. Alle Teilnehmer sind über ein Netzwerk verbunden und sehen jederzeit den gleichen Dokumentinhalt. Wenn jemand der Gruppe eine Änderung im Dokument vornimmt, wird dies in Echtzeit und synchron allen anderen Benutzern angezeigt. Jeder Benutzer hat dadurch den Überblick über alle Änderungen im Dokument. Dieser Editor ermöglicht zum Beispiel ein gemeinsames Brainstorming von mehreren Personen, welche sich an verschiedenen Orten befinden.

1.1 Zweck des Dokumentes

Das vorliegende Dokument beschreibt die Ziele, welche mit der angestrebten Lösung zu erreichen sind sowie die Anforderungen und Wünsche an das zukünftige System.

2 Ausgangslage

Das gemeinsame Entwerfen eines elektronischen Dokumentes (z.B. Textdatei), wobei jeder Beteiligte mit seinem eigenen Computer arbeitet, ist noch praktisch unbekannt. Diese Technologie birgt ein grosses Anwendungspotenzial. Der zu entwickelnde Editor ACE soll komplett neuartige und effiziente Editier-Möglichkeiten bieten und dadurch neue Wege der Zusammenarbeit eröffnen. Bis heute existiert keine marktreife, plattformunabhängige Applikation dieser Art.

3 Ist-Zustand

Die dem kollaborativen Editor zugrunde liegende Theorie entstammt aus dem Forschungsgebiet der "Computer Supported Cooperative Work - CSCW". Applikationen aus diesem Bereich werden auch mit "Groupware" bezeichnet. Seit Mitte der 90er Jahren sind zahlreiche Forschungsarbeiten geschrieben worden. Die meisten davon beinhalten theoretische Überlegungen, so zum Beispiel mathematische Beschreibungen oder Beweise zu Synchronisationsalgorithmen. Viel Theoriewissen wurde erarbeitet. Dieses Know-How soll nun in die konkrete Implementation eines kollaborativen Editors einfließen um daraus eine hochentwickelte, konkurrenzfähige Applikation auf den Markt zu bringen.

4 Ziele

In der Semesterarbeit soll die Basis gelegt werden für die Implementation eines kollaborativen und plattformunabhängigen Texteditors im Rahmen der Diplomarbeit.

Beschreibung	Priorität
Aufbau von Knowhow im Bereich CSCW	1
Bestehende Algorithmen evaluieren	1
Algorithmus implementieren	1
Testframework für Algorithmus	1
Konzept GUI	2
Konzept Netzwerk/Kommunikation	2
User Stories für kollaborativen Texteditor	3

4.1 Aufbau von Knowhow im Bereich CSCW

Es geht darum, ein fundiertes Basiswissen im Bereich des CSCW aufzubauen. Das Ziel ist, eine Übersicht über den aktuellen Forschungsstand und über die wichtigsten Errungenschaften in diesem Gebiet zu gewinnen. Das angeeignete Know-How soll beim Evaluieren der bestehenden Synchronisationsalgorithmen zum tieferen Verständnis beitragen und ein sachgerechtes Beurteilen ermöglichen.

4.2 Bestehende Algorithmen evaluieren

Die Forschung hat seit anfangs der 90er Jahre zahlreiche Synchronisations-Algorithmen in formaler Form entwickelt. Das Ziel ist eine Übersicht zu erarbeiten über die besten bis heute entwickelten Algorithmen. Jeder evaluierte Algorithmus soll prinzipiell verstanden werden. Die Übersicht soll Vor- und Nachteile aufzeigen und es ermöglichen, den am besten geeigneten Algorithmus für einen kollaborativen Texteditor zu bestimmen.

4.3 Algorithmus implementieren

Nach Evaluation eines bestimmten Algorithmus soll dieser implementiert werden (in Form einer Synchronisationsengine). Die Implementation beinhaltet ebenfalls, mögliche Nachteile respektive Schwachstellen des gewählten Algorithmus zu beheben durch eigens entwickelten Lösungen oder in Kombination mit anderen Algorithmen. Die Implementation des Algorithmus wird mit dem parallel erstellten Testframework geprüft.

4.4 Testframework für Algorithmus

Parallel zur Entwicklung des Synchronisationsalgorithmus soll ein Testframework erstellt werden. Dieses soll das sorgfältige Austesten des Algorithmus mit verschiedenen, klar definierbaren (z.B. mit Hilfe von XML-Dateien) Abläufen ermöglichen.

4.5 Konzept GUI

In erster Linie geht es darum, zu evaluieren was mit den verschiedenen Textkomponenten von Java alles möglich ist. Es muss möglich sein, mehrere Cursor sowie auswählbare und verschieden eingefärbte Textstellen in einer solchen Textkomponente darzustellen. Weiter müssen spezielle Auflistungskomponenten analysiert werden, mit welchen alle aktiven Benutzer, deren freigegebenen Dokumente und Zugriffsrechte, angezeigt werden können. Die erwähnten Komponenten sollten einfach und intuitiv zu bedienen sein.

4.6 Konzept Netzwerk/Kommunikation

Die verschiedenen Kommunikationsmechanismen, im Besonderen das gegenseitige Auffinden anderer Netzwerkteilnehmer nach dem Rendezvous-Prinzip, sollen evaluiert und verglichen werden.

4.7 User Stories für kollaborativen Texteditor

Das Ziel ist, eine Sammlung von User Stories (Anwendungsmöglichkeiten aus Sicht des Benutzers) für einen kollaborativen Texteditor zu erstellen. Diese Sammlung bildet dann eine Ideen-Grundlage für die Implementation von Funktionen des in der Diplomarbeit zu entwickelnden Texteditors.

5 Anforderungen

5.1 Architektur

Die Architektur wird basierend auf dem Peer-to-Peer Modell entwickelt. Die Applikation soll dadurch ohne einen zentralen Server funktionieren und eine einfache Installation ermöglichen.

5.2 Schnittstellen

Durch die Offenlegung des internen Kommunikations-Protokolls von ACE haben andere Applikationsentwickler die Möglichkeit, direkt mit dem Editor zu kommunizieren. Dadurch entstehen beliebige Erweiterungsmöglichkeiten mit dem kollaborativen Texteditor.