



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Seminararbeit

Benjamin Kirstgen

Home Office 2.0 - Diskursverfolgung

Benjamin Kirstgen
Home Office 2.0 - Diskursverfolgung

Seminararbeit eingereicht im Rahmen der Ringvorlesung des Master
Informatik
im Studiengang Informatik
am Department Informatik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Professor : Prof. Dr. rer. nat. Kai von Luck
Zweitgutachter : Prof. Dr. rer. nat. Gunther Klemke

Abgegeben am 28. Februar 2011

Inhaltsverzeichnis

1	Einführung	4
1.1	Computer-Supported-Cooperative-Work (CSCW)	4
1.2	Enterprise 2.0	5
2	Projekt HomeOffice 2.0	6
3	Problematik und Vision	8
3.1	Problematik	8
3.2	Vision	9
4	Masterarbeit	10
4.1	Ziel	10
4.1.1	Szenario <i>Old Software Never Dies</i>	11
4.1.2	Szenario <i>peripheral awareness</i>	11
4.2	Recherche	12
4.3	Implementierung	13
5	Ausblick/Fazit	15
	Literaturverzeichnis	16

1 Einführung

Die Möglichkeiten der Teamarbeit eines Softwareentwicklers haben sich durch die starke Vernetzung in den letzten Jahren sowie die zunehmende Bedeutung von sozialen Netzwerken als Kommunikations- und Kollaborationswerkzeuge stark verändert. Die mittlerweile standardmäßig verwendeten Werkzeuge von Softwareentwicklungsteams in Unternehmen gehen weit über einfache Intranets hinaus. Allerdings handelt es sich bei den so genannten *Enterprise 2.0*-Anwendungen häufig um Insellösungen, wodurch die Daten und das wertvolle Wissen, das sie enthalten, im Netzwerk verstreut sind. Diese Thematik wird im Projekt *HomeOffice 2.0* der HAW Hamburg behandelt.

Die vorliegende Arbeit beschreibt zum einen den Hintergrund zur Entwicklung der Masterthematik *Diskursverfolgung*, als Teil dieses Projekts und zum anderen das geplante weitere Vorgehen zur Bearbeitung der Aufgabenstellung. Dazu werden in diesem Kapitel zunächst, als Grundlage für die folgenden, einige Vorarbeiten im Bezug auf Kollaboration und Kooperation in Erinnerung gerufen. Kapitel 2 erläutert die Visionen des Projekts *HomeOffice 2.0*, um die Basis der Entscheidungsfindung zum Masterthema darzulegen. Danach werden in Kapitel 3 die spezielle Problematik sowie die daraus resultierende Vision, denen sich dieses Teilprojekt widmet, erörtert. Darauf basierend werden in Kapitel 4 das Ziel der Masterarbeit sowie das weitere Vorgehen zu deren Realisierung konkretisiert. Zum Abschluss gibt Kapitel 5 ein Ausblick auf die identifizierten Herausforderungen und die nächsten Schritte.

In den vorangegangenen Semestern wurde bereits verschiedene Thematiken des Projekts *HomeOffice 2.0* bearbeitet (siehe [13] & [14]), von denen zwei Aspekte im Folgenden kurz wiederholt werden.

1.1 Computer-Supported-Cooperative-Work (CSCW)

Bei CSCW handelt es sich um ein Forschungsgebiet, das sich mit der Computerunterstützung von kooperativem Arbeiten beschäftigt [2].

[die CSCW Forschung]...untersucht wie Gruppenmitglieder miteinander arbeiten und sucht nach Technologien sie dabei zu unterstützen... [5, S. 39]

Die Forschung beschäftigt sich demnach mit der Mensch-(Computer)-Mensch Interaktion [6]. Im Vordergrund steht allerdings die Mensch-Mensch-Interaktion. Die Bedeutung des Computers sollte möglichst transparent gestaltet werden, um ein Gefühl der direkten Zusammenarbeit zu erzeugen. [15]

Das Ziel der CSCW-Forschung besteht darin die Zusammenarbeiten von Menschen durch den Einsatz von Informations- und Kommunikationstechniken zu verbessern. Daher ist der Ausgangspunkt der Forschung immer der Mensch und wie er mit anderen interagiert. [2]

1.2 Enterprise 2.0

Ein weiteres relevantes Thema ist die neue Entwicklung der *Enterprise 2.0*-Systeme. In 2006 definiert McAfee *Enterprise 2.0* als Synonym für *enterprise social software* [18]. Darunter wird der Einsatz von sozialer Software zur Kooperation und zur Stärkung der sozialen Bindung im Unternehmen verstanden [18].

Auch bei diesen Anwendungen steht der Mensch im Vordergrund. Daher soll die Technologie den Gewohnheiten und Bedürfnissen der Mitarbeiter angepasst werden - und nicht etwa umgekehrt.

2 Projekt HomeOffice 2.0

Das Projekt *HomeOffice 2.0* wurde im Rahmen des Masterstudiengangs an der HAW-Hamburg im Wintersemester 2009/10 entwickelt [16]. Im Folgenden sollen dessen Visionen erläutert werden, um damit die Grundlage für die Wahl der Masterthematik darzulegen, die in der dieser Arbeit vorgestellt wird.

- Entwickler bei Teamarbeit** – Das primäre Ziel des Projekts *HomeOffice 2.0* ist die Unterstützung von Softwareentwicklern bei ihrer Arbeit in verschiedenen Teams und Projekten. Dabei sollen spezifische Informationen zu den Projekten gesammelt und strukturiert verwaltet werden, so dass dem Entwickler sowohl der Wechsel zwischen Projekten während der Arbeit, als auch der Zugriff auf ältere Informationen erleichtert wird. Durch Kommunikations- und Kollaborationswerkzeuge soll eine virtuelle Nähe zwischen Teammitgliedern geschaffen und deren Bewusstsein für einander gestärkt werden.
- Mobiler Arbeitsplatz** – An die Unterstützung des Entwicklers bei der Teamarbeit knüpft eine weitere Vision, die Gestaltung eines mobilen Arbeitsplatzes, an. Dabei soll es dem Entwickler erleichtert werden von verschiedenen Standorten aus (Büro, zu Hause, unterwegs) an der Projektarbeit teilzunehmen und in den Projektkontext eingebunden zu sein.
- Wissensmanagement** – Ein weiterer wichtiger Aspekt des Projektes *HomeOffice 2.0* ist die Unterstützung des Wissensmanagements in Unternehmen. Hierbei gilt es das Wissen und das KnowHow der einzelnen Wissensarbeiter eines Unternehmens strukturiert zu verwalten. Dies ermöglicht einen besseren Zugang zu dem gesammelten Unternehmenswissen und macht die vorhandenen Kompetenzen sichtbar (siehe Abbildung 2.1). [16]

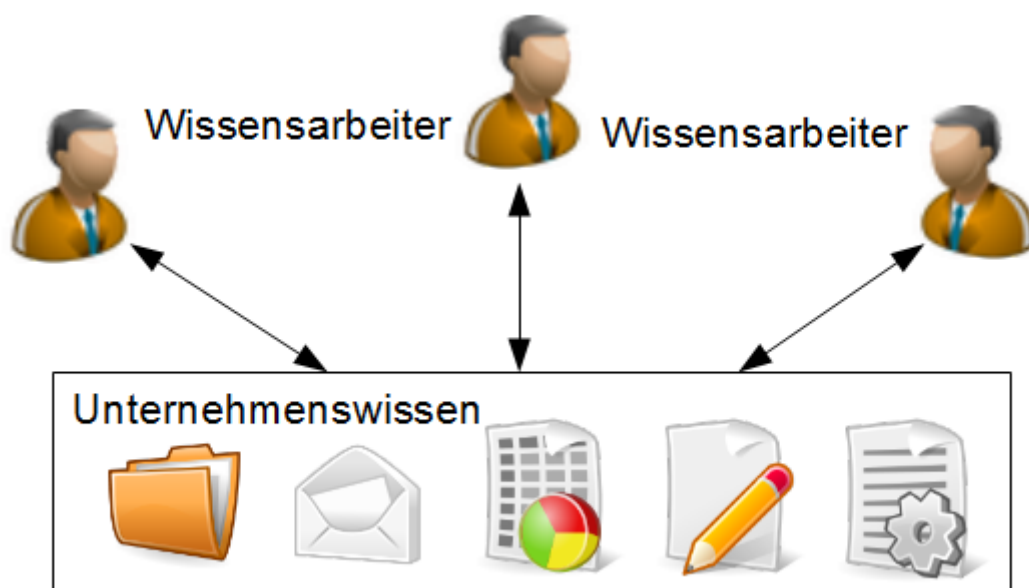


Abbildung 2.1: Wissensmanagement in Unternehmen

3 Problematik und Vision

Anlehnend an die Visionen des Projekts *HomeOffice 2.0*, wird im Folgenden die Problematik beschrieben, der sich das Teilprojekt *Diskursverfolgung* widmet (Kap. 3.1). Anschließend wird die Vision vorgestellt, die dieser entgegen wirken soll (Kap. 3.2).

3.1 Problematik

Bei der Softwareentwicklung gibt es einige Faktoren, die das Arbeiten erschweren:

- Mehrere Projekte** – Ein Entwickler ist meistens in mehreren Projekten gleichzeitig aktiv und muss an einem Tag häufiger zwischen diesen Wechseln.
- Verschiedene Rollen** – In den verschiedenen Projekten nimmt ein Mitarbeiter oft auch unterschiedliche Rollen an. So kann er bspw. in einem Projekt als Entwickler und in einem anderen als Projektleiter tätig sein. Dabei sind je nach Rolle auch unterschiedliche Informationen eines Projektes von Relevanz.
- Old Software Never Dies** – Auch wenn Software, im Gegensatz zur Hardware, keinen Verschleiß aufweist, muss sie dennoch häufig in der Produktion gewartet werden. So führen etwa neue Umgebungsfaktoren dazu, dass Software angepasst werden muss. Ein bekanntes Beispiel hierfür war die Jahrtausendumstellung. [16]

Durch diese verschiedenen Faktoren wird ein Wechsel zwischen Projekten häufiger notwendig. Dies kostet den Mitarbeiter viel Zeit, da er sich jedes Mal den Kontext des Projekts und der Arbeitsaufgabe wieder aufbauen muss. [11]

Aus der Flut der im Unternehmen vorhandenen Informationen müssen bei jedem Wechsel die jeweils passenden herausgefiltert werden. Diese Problematik wird als *kognitive Überlastung* bezeichnet und kann in Stresssituationen resultieren, die letztlich auch die Produktivität einschränken.[12]

3.2 Vision

Die geschilderte Problematik soll im Rahmen der Masterarbeit *Diskursverfolgung* behandelt werden. Dabei wird die Vision verfolgt den Wechsel zwischen Projekten oder den Neueinstieg zu vereinfachen. Dies soll mit Hilfe einer strukturierten Darstellung des Diskurses einer Ressource erreicht werden.

Die aktuelle Arbeitsaufgabe soll dadurch unterstützt werden, dass alle Informationen, die für die jeweilige Ressource relevant sind, angezeigt werden können. Um dies zu realisieren, soll die Entwicklung an einer Ressource in der Entwicklungsumgebung sichtbar gemacht und der Diskurs, der währenddessen erstellt wurde, dargestellt werden (siehe Abbildung 3.1). Der Zugriff auf diese Informationen soll möglichst intuitiv und übersichtlich aus der IDE heraus gestaltet werden.

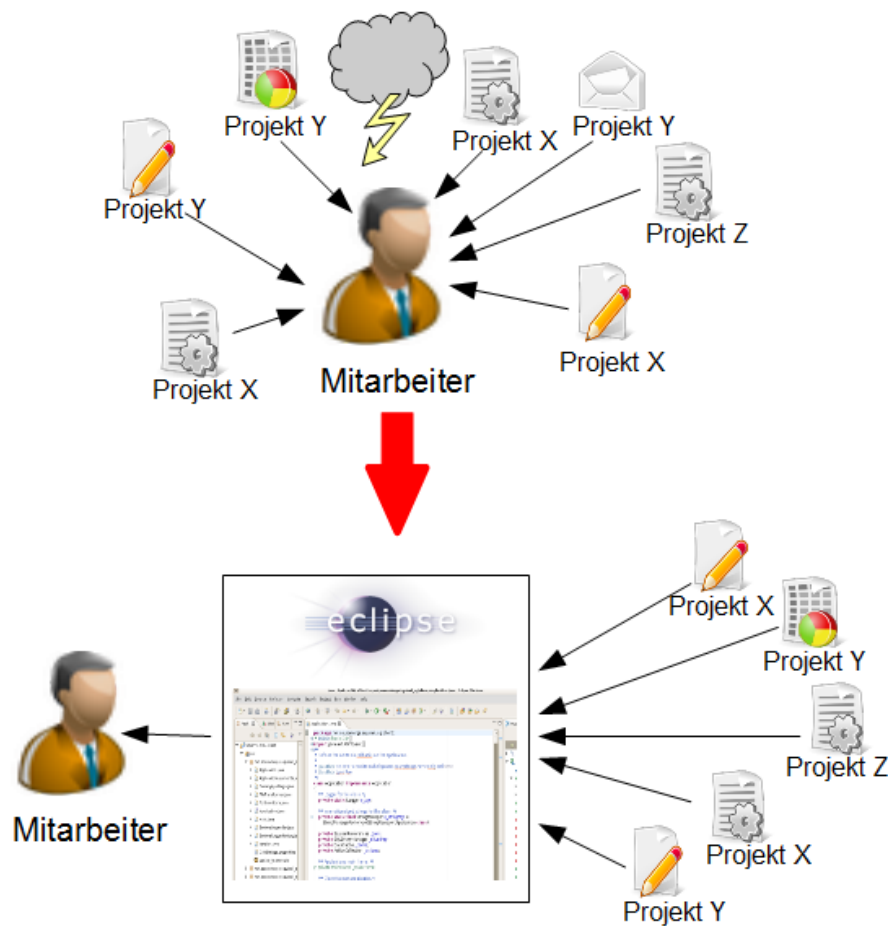


Abbildung 3.1: Vision der *Diskursverfolgung*

4 Masterarbeit

In diesem Kapitel wird die Masterarbeit *Diskursverfolgung* vorgestellt. Dazu wird zunächst das Ziel der Arbeit spezifiziert, um den Nutzen eines solchen Projekts darzulegen. Dieser wird anhand zweier Szenarien demonstriert. Anschließend wird das weitere Vorgehen zur Bearbeitung der Aufgabenstellung erläutert. Dies erfolgt zum einen bezüglich der Recherche, zum anderen hinsichtlich der Implementierung.

4.1 Ziel

Aus der in Kapitel 3.2 vorgestellten Vision leitet sich das Ziel der Masterarbeit *Diskursverfolgung* ab. Der Entwickler soll bei seiner Arbeit unterstützt werden, indem relevante Informationen über die aktuell zu bearbeitende Ressource aggregiert dargestellt werden. Dazu werden dateispezifische Informationen aus verschiedenen Entwicklungswerkzeugen (Wiki, Versionsverwaltung, Ticketsystemen) zusammengeführt. Aus diesen wird ein Diskurs erstellt, der den Entwicklungsprozess der Ressource übersichtlich darstellt. Dabei wird nicht nur die Entwicklung des Codes über die Zeit berücksichtigt, sondern auch zusätzliche Informationen zu Wiki-seiten, Arbeitspaketen, vorherigen Bearbeitern und Tickets hervorgehoben, die währenddessen erstellt wurden. Dieser Diskurs soll so realisiert werden, dass er für den Entwickler möglichst übersichtlich und bedarfsorientiert aus der Entwicklungsumgebung heraus zugänglich ist.

In einem agilen Softwareentwicklungsteam mit *weak* oder *collectiv Code ownership* [8] ist nicht nur der vergangene Diskurs einer Ressource von Interesse, sondern auch deren aktuelle Entwicklung. Daher wird das Werkzeug der *Diskursverfolgung* als *peripheral awareness*-System realisiert, um dem Entwickler zusätzlich Informationen über aktuelle Entwicklungen an der Ressource und mögliche Konflikte bereit zu stellen.

Bei der Entwicklung von *peripheral awareness*-Systemen wird die Verwendung der unterbewussten Wahrnehmung gefördert, um das Bewusstsein nur geringfügig von der aktuellen Aufgabe abzulenken [9]. Der Aufwand zur Aufnahme der Informationen sollte für den Benutzer so gering wie möglich gehalten werden [10].

Die Anzeige der *peripheral awareness* in der Entwicklungsumgebung lehnt sich an die Darstellungsform bekannter IM (Instant Messaging)- Systeme [22] an. Allerdings wird nicht der Status anderer Benutzer angezeigt, sondern die aktuellen Aktivitäten an einer Ressource signalisiert.

Um den Nutzen eines solchen Werkzeugs zu verdeutlichen, werden im Folgenden zwei Szenarien vorgestellt:

4.1.1 Szenario *Old Software Never Dies*

In diesem Szenario (siehe Abbildung 4.1) wird die Bearbeitung eines Fehlers an einer Klasse des bereits abgeschlossenen Projektes KV-Bestand dargestellt. Das Projekt wurde zur Jahrtausendumstellung bereits wiedereröffnet. Allerdings handelt es sich bei dem Bearbeiter, der den Fehler beheben soll, um einen neuen Mitarbeiter. Dieser muss sich erst in das Projekt einarbeiten. Die *Diskursverfolgung* soll ihn dabei unterstützen, indem der Diskurs der zu bearbeitenden Klasse aus der Entwicklungsumgebung abrufbar ist. Dieser beinhaltet unter anderem Informationen zu vorherigen Bearbeitern, Wikiseiten mit relevanten Daten und mit der Klasse verbundene Arbeitsaufgaben. Dadurch kann sich der Mitarbeiter besser den Kontext der aktuellen Klassen aufbauen und schneller mit der effektiven Fehlerbehebung beginnen.

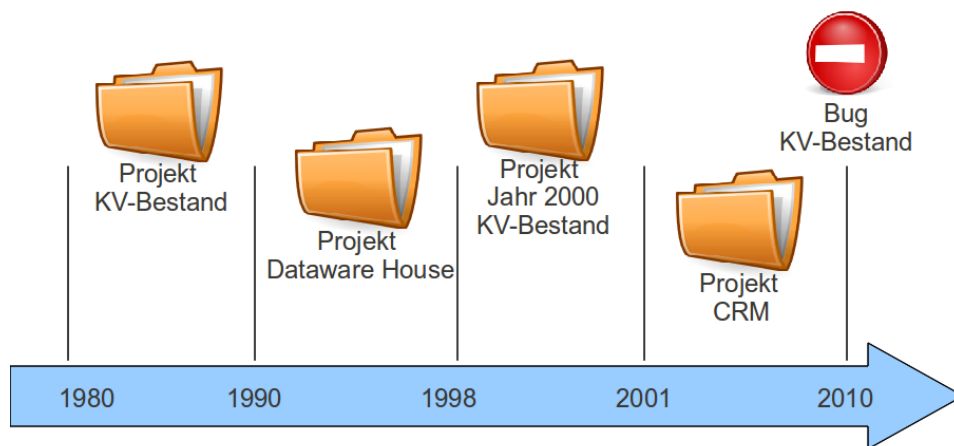


Abbildung 4.1: Szenario *Old Software Never Dies*

4.1.2 Szenario *peripheral awareness*

Ein Mitarbeiter erhält den Arbeitsauftrag eine Methode zu einer bestehenden Klasse hinzuzufügen. Allerdings ist die Arbeit an der Klasse noch nicht abgeschlossen und sie wird von verschiedenen Abteilungen verwendet und modifiziert. Daher ist es möglich, dass auch andere Mitarbeiter gleichzeitig oder zeitnah an der Ressource arbeiten. Das *peripheral awareness*-System soll dem Bearbeiter signalisieren, ob dies der Fall ist. So kann beispielsweise an einer

Ampel-Darstellung erkannt werden wie intensiv die Arbeiten an einer Klasse zu diesem Zeitpunkt sind. Dadurch können mögliche Konflikte und Überschneidungen frühzeitig angezeigt und umgangen werden.

4.2 Recherche

Während der Erarbeitung des Masterthemas *Diskursverfolgung* haben sich einige Punkte herauskristallisiert, die eine weiterführende Recherche erfordern. Dazu gehört die Suche nach Produkten oder Projekten, die sich mit einer ähnlichen Problemstellung beschäftigen und wie deren Informationen und/oder Implementierungen unter Umständen genutzt werden können. Dabei haben sich bisher vor allem zwei Werkzeuge als besonders interessant hervorgehoben:

- Mylyn**
 - Bei Mylyn handelt es sich um ein Eclipse Plug-In, das eine aufgaben-fokussierte Benutzeroberfläche bietet. Es unterstützt den Entwickler, indem die Informationsflut in der Entwicklungsumgebung reduziert wird. Um dies zu erreichen werden nur jene Ressourcen in der IDE angezeigt, die für die aktuelle Arbeitsaufgabe relevant sind. Mylyn registriert während der Tätigkeit des Entwicklers an einer Aufgabe alle verwendeten Ressourcen und erstellt so einen Kontext der aktuellen Arbeitsaufgabe. Dadurch können bei ihrem nächsten Start nur die relevanten Ressourcen angezeigt werden. [19] Für die *Diskursverfolgung* ist dabei vor allem der erstellte Kontext interessant und wie er genutzt werden kann um eine *peripheral awareness* zu schaffen.
- Tasktop**
 - Bei Tasktop handelt es sich um ein proprietäres Produkt das ebenfalls einen Aufgabenkontext erstellt. Allerdings handelt es sich bei Tasktop um eine Desktopanwendung, die nicht nur die Aktivitäten in der IDE registriert, sondern auch jene außerhalb der Entwicklungsumgebung, etwa den Verlauf des Browsers. [21] Da Tasktop eine Weiterentwicklung des OpenSource-Plug-Ins Mylyn ist, kann es in jede Eclipse-basierte Entwicklungsumgebung eingebunden werden [20]. Auch bei Tasktop ist es wiederum von Interesse welche Schnittstellen das Programm möglicherweise bietet und wie der erstellte Aufgabenkontext genutzt werden kann.

Weitere Recherche-Themen beschäftigen sich mit den verschiedenen Arbeitsweisen von Entwicklern und wie die *Diskursverfolgung* sie am effektivsten dabei unterstützen kann. Dazu gehört die Gestaltung der Benutzeroberfläche, also die Darstellung des Diskurses in der Entwicklungsumgebung. Außerdem muss identifiziert werden welche Informationen der verschiedenen Werkzeuge für den Diskurs einer Ressource relevant sind.

Hinsichtlich der Darstellung der aktuellen Aktivitäten an einer Ressource muss recherchiert werden, wie diese gestaltet werden sollte, damit sie den Anforderungen an ein *peripheral awareness*-System entspricht (siehe Kapitel 4.1).

4.3 Implementierung

Da es sich bei der *Diskursverfolgung* um ein Teilprojekt des Projektes *HomeOffice 2.0* handelt, wird das zu entwickelnde Werkzeug in Verbindung mit dem *LuPanKu*-Server stehen (siehe Abbildung 4.2) [16]. Auf der Abbildung ist die Eingliederung der *Diskursverfolgung* in das *LuPanKu*-System anhand der roten Markierung nachvollziehbar. In gemeinschaftlicher Arbeit wurde bereits in Projekt 1 & 2 die dargestellte Architektur, mit der Connector-Schnittstelle und der Infrastruktur zur Speicherung der Meta-Informationen realisiert. Die beiden Hauptaufgaben für die hier vorgestellte Arbeit sind nun das Erstellen eines Eclipse Plug-Ins als Benutzerschnittstelle [4] und eines Analyzers für das *LuPanKu*-System.

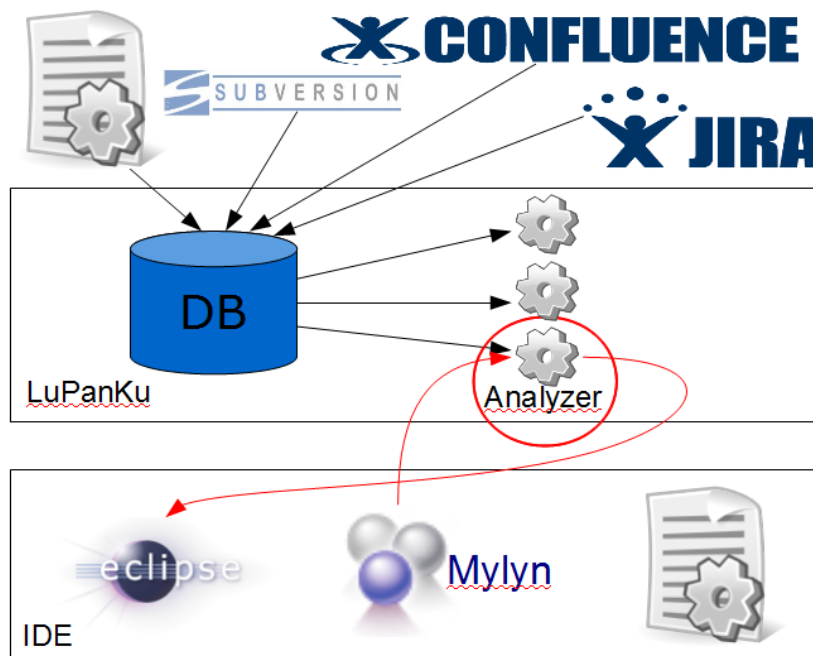


Abbildung 4.2: Architektur *LuPanKu*-System mit Anpassungen für *Diskursverfolgung*

Das Eclipse Plug-In dient der Darstellung des Diskurses und der *peripheral awareness* in der Eclipse-basierten Entwicklungsumgebung. Der Analyzer ist verantwortlich für die Aggregation der relevanten Daten aus der Fülle der Meta-Informationen. Aus diesen erstellt er zum einen den Diskurs der aktuellen Ressource und zum anderen die Informationen zur *peripheral awareness*. Diese leitet er an die IDE weiter.

Bei einem Plug-In handelt es sich um die kleinste funktionale Einheit innerhalb von Eclipse, die als eigenständige entwickelt, getestet und weitergegeben werden kann [3]. Eclipse basiert

ab der Version 3 auf dem Standard der OSGi¹ [1] und stellt ein Framework zur einfachen Integration von Plug-Ins (*bundles*) zur Verfügung [7]. Diese Architektur basiert auf einem Komponentenmodell und ermöglicht das Hinzufügen und Entfernen von Plug-Ins zur Laufzeit, ohne dass Eclipse neu gestartet werden muss [3]. Außerdem ist bei der Entwicklung von Eclipse Plug-Ins darauf zu achten, dass Eclipse das Standard Widget Toolkit (SWT) zur Darstellung aller graphischen Objekte nutzt. Deshalb müssen alle Plug-Ins das SWT verwenden, wenn sie in Eclipse integriert werden sollen.

Der Analyzer wird zur Einbindung in das *LuPanKu*-System ebenfalls als OSGi-Service [16] implementiert. Er filtert die Flut der Meta-Informationen nach relevanten Informationen für die entsprechende Ressource. Daraus erstellt er einen Diskurs, der an die Entwicklungsumgebung weitergeleitet wird. Wie im Kapitel 4.2 bereits beschrieben muss noch recherchiert werden, wie die Filter zur Aggregation konfiguriert werden müssen und wie relevante Daten erkannt werden können.

Bei der Erstellung der Informationen für das *peripheral awareness*-System kann der Analyzer vor allem den von Mylyn erstellten Aufgabenkontext verwenden. In diesem sind Informationen zu allen aktuell bearbeiteten Ressourcen der Kollegen enthalten. In Zusammenarbeit mit Karsten Panier wurde im Rahmen von Projekt 2 ein erster Prototyp entwickelt, bei dem sich Clients mit dem *LuPanKu*-Server verbunden und ihren aktuellen Mylyn-Kontext verbreitet haben. [17] Weitere Information zur Status-Anzeige können beispielsweise aus der Anzahl von Commits und der Anzahl der verschiedenen Bearbeiter einer Ressource gewonnen werden. Dies ermöglicht Konflikte frühzeitig zu ermitteln und den Entwickler vorzuwarnen. Eine gängige Möglichkeit der Darstellung der *peripheral awareness* in der Entwicklungsumgebung ist das Ampel-System, wobei Rot eine hohe Aktivität und Grün die freie Bearbeitung signalisiert.

¹Open Service Gateway Initiative

5 Ausblick/Fazit

Durch das Arbeiten an der Masterarbeit wird die bisher enge Zusammenarbeit mit den Projektmitgliedern des Projekts *HomeOffice 2.0* aufgelockert. Da das *LuPanKu*-System sehr modular konzipiert ist, kann sich jedes Mitglied mit der Entwicklung der eigenen Analyserkomponenten (siehe Abbildung 4.2) beschäftigen. Allerdings wird auf Grund des Projektrahmens und der sozialen Konstellation dennoch ein reger Austausch stattfinden. So wird beispielsweise gemeinschaftlich an dem Prototypen des *peripheral awareness*-Systems weitergearbeitet.

Der nächste Schritt beim Verfassen der Masterarbeit ist die Recherche der in Kapitel 4.2 beschriebenen Aspekte. Dazu gehört auch die Erstellung eines effektiven Diskurses, die wohl größte Herausforderung der Arbeit. Denn dieser muss aus der Flut der Informationen der *LuPanKu*-Datenbank sowie der Mylyn-Kontexte die jeweils relevanten herausfiltern können. Zusätzlich müssen nicht explizit erstellte Verbindungen zu den Meta-Informationen der Werkzeuge erkannt werden. Es wird angestrebt bereits früh einen sehr einfachen Prototypen des Analyzers und des Eclipse Plug-Ins zu realisieren, um rechtzeitig mögliche nicht-erkannte Zusammenhänge aufzudecken. Die endgültige Darstellung des Diskurses in der Entwicklungsumgebung wird erst zu einem späteren Zeitpunkt implementiert, da sie der Entwicklung des Analyzers angepasst wird. Jedoch muss dann besonders auf die Usability des Werkzeugs geachtet werden, da diese entscheidend für die Akzeptanz durch die Entwickler ist.

Zusätzlich muss beim Verfassen der Masterarbeit darauf geachtet werden, dass weitere Ideen letztlich nicht deren Rahmen sprengen, da sich bei der Realisierung häufig neue Erweiterungen und Möglichkeiten ergeben.

Literaturverzeichnis

- [1] ALLIANCE, OSGi: *OSGi*. 2011. – URL <http://www.osgi.org/Main/HomePage>. – Zugriffsdatum: 12. January 2011
- [2] BACK, Andrea: *CSCW - Workflow und Groupware*. Heidelberg : dpunkt.verlag GmbH, 2000
- [3] BOLOUR, Azad: *Notes on the Eclipse Plug-in Architecture*. 2003. – URL http://www.eclipse.org/articles/Article-Plug-in-architecture/plugin_architecture.html. – [Online; Stand 22. Februar 2011]
- [4] CLAYBERG, Eric ; RUBEL, Dan: *Eclipse: Building Commercial-Quality Plug-ins (2nd Edition)*. Addison-Wesley Professional, 2006. – ISBN 032142672X
- [5] ELLIS, S. J. Rein G. L.: Groupware Some Issues and Experiences. In: *Communications of the ACM* 34 (1991), S. 5–22
- [6] FG, CSCW: *Homepage Fachgruppe CSCW der Gesellschaft fuer Informatik*. 2009. – URL <http://www.fgcsw.de/>
- [7] FOUNDATION, Eclipse: *Eclipse Projects - OSGi*. 2011. – URL <http://www.eclipse.org/osgi/>. – Zugriffsdatum: 12. January 2011
- [8] FOWLER, Martin: *Code Ownership*. 2006. – URL <http://martinfowler.com/bliki/CodeOwnership.html>
- [9] GENNARO, Dr. Rocco J.: Representationalism, peripheral awareness, and the transparency of experience. In: *Philosophical Studies* (2007). – URL <http://www.theassc.org/files/assc/repperiph.pdf>
- [10] GROSS, Michael: *Computer-Supported Cooperative Work*. Oldenbourg Wissenschaftsverlag GmbH, 2007
- [11] KERSTEN, Mik: *Focusing knowledge work with task context*. Vancouver, BC, Canada, Canada, Dissertation, 2007
- [12] KIRSH, David: A Few Thoughts on Cognitive Overload. In: *Intellectica* 30 (2000), S. 19–51

- [13] KIRSTGEN, Benjamin: Computer Supported Collaborative Work (CSCW). (2010). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/kirstgen/bericht.pdf>. – Zugriffsdatum: 22. February 2011
- [14] KIRSTGEN, Benjamin: Schatten IT und Beispiele für Kollaboration in Enterprise 2.0. (2010). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-aw2/kirstgen/bericht.pdf>. – Zugriffsdatum: 22. February 2011
- [15] KOECKRITZ, Oliver: Geschichte und Konzepte von Collaborative Workspaces. In: *Seminararbeit im Fach AWI* (2006). – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2006/koeckritz/abstract.pdf>
- [16] MATTHIAS HOLSTEN, Karsten P.: *Projektbericht Home-Office 2.0*. 2010. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master2010-proj1/panier.pdf>
- [17] MATTHIAS HOLSTEN, Karsten P.: *Projektbericht Home-Office 2.0 - Second Milestone*. 2011
- [18] MCAFEE, Andrew P.: Enterprise 2.0: The Dawn of Emergent Collaboration. In: *MITS-loan* 47 (2006), Spring, Nr. 3, S. 21–28
- [19] TEAM, Mylyn: *Mylyn / User Guide*. Eclipse Fondation. – URL http://wiki.eclipse.org/index.php/Mylyn/User_Guide
- [20] TEAM, Tasktop: *Tasktop - Mylyn Code at the speed of thought*. 2011. – URL <http://tasktop.com/mylyn/>. – [Online; Stand 22. Februar 2011]
- [21] TEAM, Tasktop: *Tasktop - Task-focused productivity software built on Eclipse Mylyn*. 2011. – URL <http://tasktop.com/>. – [Online; Stand 22. Februar 2011]
- [22] WIKIPEDIA: *Instant Messaging — Wikipedia, Die freie Enzyklopädie*. 2011. – URL http://de.wikipedia.org/w/index.php?title=Instant_Messaging&oldid=83733164. – [Online; Stand 6. Februar 2011]

Abbildungsverzeichnis

2.1	Wissensmanagement in Unternehmen	7
3.1	Vision der <i>Diskursverfolgung</i>	9
4.1	Szenario <i>Old Software Never Dies</i>	11
4.2	Architektur <i>LuPanKu</i> -System mit Anpassungen für <i>Diskursverfolgung</i>	13