



Hochschule für Angewandte Wissenschaften Hamburg
Hamburg University of Applied Sciences

Home-Office2.0 Related Work

Karsten Panier

*Fakultät Technik und Informatik
Department Informatik*

*Faculty of Engineering and Computer
Science
Department of Computer Science*

Inhaltsverzeichnis

1	Einleitung	3
2	Related Work	4
2.1	ACTIVE Project	4
2.1.1	Information Delivery	5
2.1.2	Bezug zu Home Office 2.0	6
2.2	Task Context	6
2.2.1	User Interaction History	7
2.2.2	Degree Of Interest	8
2.2.3	Kontext Operationen	8
2.2.4	Architektur	9
2.2.5	Bezug zu Home Office 2.0	10
2.3	Socio-Technical Congruence	10
2.3.1	Socio-Technical Network	11
2.3.2	Bezug zu Home Office 2.0	12
3	Zusammenfassung	13
	Literaturverzeichnis	14

1 Einleitung

Die technologischen Entwicklungen der Informationssysteme haben die Arbeitsplätze von Wissensarbeitern grundlegend verändert. In dieser Arbeit werden Projekte diskutiert, die einen Beitrag zur Unterstützung der modernen Wissensarbeit leisten. Zudem werden diese in Beziehung zu Home Office 2.0 [19] gesetzt und auf ihren Nutzen überprüft.

Das Projekt Home Office 2.0 ist mit seiner Vision und den daraus abgeleiteten Anforderungen in [19] beschrieben. Nachfolgend werden die Themenschwerpunkte von Home Office 2.0 zusammengefasst.

- Wissensarbeit
- Softwareentwicklung
- Den Diskurs nachvollziehen
- Informationsflut vermeiden
- Beim Wechsel zwischen den Aufgaben unterstützen
- Zusammenarbeit
- Verteilte Teams
- Informationsbeschaffung

Die vorgestellten Themenschwerpunkte für Home Office 2.0 adressieren den Wissensarbeiter als Nutzer. Deshalb wird auf diese der Fokus gesetzt und nicht auf die Geschäftsprozesse oder die technische Infrastruktur. Damit folgt Home Office 2.0 dem gleichen Ansatz wie die Agile Softwareentwicklung, in welcher Individuen über Prozesse und Werkzeuge gestellt werden [8]. Auch das Thema Enterprise 2.0 stellt den Menschen in den Vordergrund. Hier wird allerdings mehr Wert auf die soziale Interaktion der Mitarbeiter gelegt, um relevante Lösungen zu erzeugen [18].

2 Related Work

Bevor auf die einzelnen Projekte eingegangen wird, werden die Tätigkeiten eines Wissensarbeiters kurz vorgestellt.

- Relevante Informationen finden
Die Suche nach relevanten Informationen in einer großen Datenmenge, kann beliebig zeitaufwändig sein [17].
- Erzeugen und modifizieren von Wissen
Zur Lösung von konkreten Aufgaben muss bestehendes Wissen modifiziert oder neues Wissen entwickelt werden [22].
- Lernen
Durch die kontinuierliche Weiterentwicklung von Wissen, muss der Wissensarbeiter sich ständig weiterbilden [11].
- Wissen vermitteln
Die Erkenntnisse und Prozessänderungen des Wissensarbeiters muss er an seine Kollegen weitergeben [11].
- Aufgaben-orientierte Arbeitsweise
Die Arbeitsweise eines Wissensarbeiters ist durch das Erledigen von Aufgaben definiert und nicht durch einen formalen Prozess [14].

Nachfolgend werden Projekte vorgestellt, die zum einen den Prozess der Wissensarbeit betrachten und zum anderen konkrete Unterstützung bei der Nutzung von Werkzeugen anbieten.

2.1 ACTIVE Project

Das EU-Projekt ACTIVE [3] ist ein Konsortium, bestehend aus zwölf Organisationen aus sieben europäischen Ländern. Koordiniert wird es von der britischen Telekom. Die Zielsetzung des Projektes ist es, die Produktivität von Wissensarbeitern in einer proaktiven und kontextbezogenen Art zu verbessern.

Das Projekt verfolgt drei Forschungsschwerpunkte:

- Einfacher Informationsaustausch
- Teilen und Wiederverwenden von Wissensprozessen
- Verstehen des Benutzerkontextes

2.1.1 Information Delivery

ACTIVE verfolgt das Ziel, Informationen passend für eine Aufgabe bereitzustellen und diese mit anderen zu teilen. Dafür wird der Ansatz einer kontextsensitiven Arbeitsumgebung [21] verfolgt.

Bei der Analyse der Kontexte, in denen ein Wissensarbeiter seine Aufgabe erledigt, wurden sechs Aspekte herausgearbeitet:

- Informell
Welche Dokumente wurden während der Bearbeitung einer Aufgabe gelesen, modifiziert oder angelegt.
- Organisatorisch
Welche Organisationseinheiten sind von einer Aufgabe betroffen oder müssen zu ihrer Bearbeitung involviert werden.
- Verhalten
Welche Aktionen führt der Benutzer bei der Bearbeitung einer Aufgabe aus.
- Operational
Welche Werkzeuge und Anwendungen werden benutzt.
- Kausal
Welches Ziel hat die Aufgabe.
- Chronologisch
In welcher zeitlichen Reihenfolge werden Interaktionen mit den Werkzeugen ausgeführt.

Das ACTIVE Projekt identifiziert die Wissensarbeit als einen informellen Prozess. Die Bearbeitung einer Aufgabe erfolgt in mehreren Schritten, die insgesamt einen Prozess bilden. Zu Anfang ist dieser nicht definiert und entwickelt sich während der Bearbeitung der Aufgabe. Er besteht aus Aktionen, die zu neuen Erkenntnissen führen. Durch diese ergeben sich Entscheidungen, die wiederum zu neuen Aktionen führen.

Dieser flexible und adaptierbare Prozess der Wissensarbeit ist eingebettet in den Geschäftsprozess des Unternehmens. Aus dieser Beziehung leitet ACTIVE vier Kontexte ab, in denen ein Wissensprozess gestartet werden kann:

- Arbeits-Kontext
Ein Prozess der Wissensarbeit wird hierbei als Bestandteil der Arbeit gesehen.
- Geschäfts-Kontext
Ein formaler Geschäftsprozess wird ausgeführt und benötigt neues Wissen.
- Aufgaben-Kontext
Ein Wissensprozess wird durch einen anderen Wissensprozess angestoßen.
- Umgebungs-Kontext
Ein Wissensprozess kann durch ein unerwartetes Ereignis von außen ausgelöst werden.

2 Related Work

Diese Kontexte für die Wissensarbeit können kombiniert werden. ACTIVE bildet den Kontext aus Dateien oder Emails, die der Benutzer häufig oder in einer kurzen Zeitspanne aufgerufen hat. Eine Unterstützung von Teams wird möglich, wenn die Dateien in einem gemeinsam genutzten Repository liegen. Es werden dann die Kontexte der Benutzer miteinander verglichen. Wenn sich zwei Kontexte ähnlich sind, können sie zu einem gemeinsamen zusammengefasst werden. Dadurch erhalten beide Benutzer Informationen über Dokumente, die für ihre Arbeit eine Rolle spielen können. Der Kontext wird herangezogen, um dem Benutzer in verschiedenen Dialogen Dokumentenvorschläge anzubieten. [14]

2.1.2 Bezug zu Home Office 2.0

Das ACTIVE Projekt stellt einen Bezug zwischen dem informellen Prozess der Wissensarbeit und Geschäftsprozessen her. Diese Betrachtung sollte im Projekt Home Office 2.0 berücksichtigt werden.

Der Ansatz ein Kontext-Repository zu verwenden, um gezielt Informationen und Prozesse im Team zu teilen, wird für das Projekt Home Office 2.0 übernommen. Es unterstützt die Verfolgung des Diskurses und kann den Mitarbeitern helfen, sich trotz unterschiedlicher Standorte den aktuellen Tätigkeiten der Kollegen bewusst zu sein.

Zur Verifikation der vorgestellten Konzepte fehlt noch eine Überprüfung durch eine Integration in Werkzeuge. Diese ist momentan nur in Grundzügen realisiert und nicht frei verfügbar [14].

2.2 Task Context

Die an der Universität von British Columbia angefertigte Doktorarbeit von Mik Kersten beschäftigt sich mit dem Kontext des Softwareentwicklers bei der Bearbeitung einer Aufgabe. Die Arbeit *Focusing knowledge work with task context* adressiert dabei zwei konkrete Probleme von Softwareentwicklern:

- Information overload problem
Werden Informationen aufgrund des ihnen zugrunde liegenden Systems dargestellt, sind es mehr Informationen als für die aktuelle Bearbeitung einer Aufgabe benötigt werden. Dies führt zu Verwirrung und erschwert das Auffinden von gewünschten Informationen. [17] [16]
- Context loss problem
Bei der Bearbeitung einer Aufgabe werden Informationen gesammelt und Wissen über die Lösung aufgebaut. Wird die Arbeit unterbrochen, um beispielsweise eine andere Aufgabe zu erledigen, geht dieses Wissen verloren. [15] [16]

2 Related Work

Die Arbeit konzentriert sich auf die Reduzierung dieser Probleme in Entwicklungsumgebungen. Dafür entwickelte Mik Kersten das Plug-In Mylyn [2] für die Eclipse IDE [13]. Zunächst wurden die gängigen Navigationsmöglichkeiten in Entwicklungsumgebungen betrachtet, um die relevanten Informationen über das zu entwickelnde System zu bekommen:

- Hierarchische Sichten
- Inhaltsverzeichnisse
- Suchmaschinen
- Markierungen und Bookmarks
- Strukturabhängigkeiten im Quellcode
 - Referenzen auf andere Objekte
 - Suche nach Objekten, die ein bestimmtes Objekt verwenden
 - Navigation innerhalb der Vererbungshierarchie

Bei diesen Navigationsformen muss der Entwickler sich ständig in einer großen Menge von Objekten bewegen. Besonders die Suche in hierarchischen Sichten und Inhaltsverzeichnissen wird aufwändig, da dort keine kontextabhängigen Filter verfügbar sind. Lediglich bei Suchdialogen kann die Ergebnismenge eingegrenzt werden.

Der Ansatz von Mylyn ist die Bereitstellung von Filtern für die verschiedenen Sichten, die die benötigten Ressourcen hervorheben. In Sichten, die generell alle Informationen anzeigen, werden die Notwendigen besonders herausgestellt und Irrelevante ausgeblendet. In Suchmaschinen werden die Ergebnisse in einer Rangliste dargestellt. Gewonnen werden die dafür notwendigen Informationen aus dem Kontext der aktuellen Aufgabe des Entwicklers.

Das *Context loss problem* wird durch die Speicherung des Kontextes zu einer Aufgabe reduziert. Muss ein Entwickler seine aktuelle Tätigkeit unterbrechen und eine andere Aufgabe erledigen, speichert Mylyn den aktuellen Status der Entwicklungsumgebung. Bei der Aktivierung der vorherigen Tätigkeit stellt Mylyn diesen wieder her. Dies soll es dem Entwickler erleichtern, sich wieder in den Kontext der Aufgabe einzufinden.

2.2.1 User Interaction History

Der Kontext wird von Mylyn durch das Verfolgen der Interaktionen des Benutzers mit der Entwicklungsumgebung hergestellt. Dafür wählt der Entwickler zunächst eine Aufgabe aus. Die Artefakte, die er dann benutzt, werden mit der Aufgabe assoziiert und ergeben die *Interaction History*. Die *Interaction History* ist eine Menge an Benutzerereignissen mit folgenden Eigenschaften:

- Zeitpunkt des Ereignisses

- Art des Ereignisses
 - Direkte Interaktion
 - * Selektion
 - * Bearbeiten
 - * Befehl
 - Indirekte Interaktion
 - * Propagieren
 - * Prognose
- Typ des betroffenen Objektes
- Referenz auf das Objekt

2.2.2 Degree Of Interest

Für die Transformation von der *Interaction History* zu einem Aufgaben-Kontext wird der Degree-Of-Interest-Algorithmus (DOI) verwendet. Dieser Algorithmus ermittelt das Interesse des Benutzers an den Objekten, mit denen er gearbeitet hat. Dafür werden die Benutzerereignisse anhand der Objekte gruppiert. Der Grad des Interesses wird zum einen an der Menge der Ereignisse zu einem Objekt gemessen und zum anderen wie lange das Ereignis zurückliegt. Daher sinkt der ermittelte Grad des Interesses an Objekten, wenn sie längere Zeit nicht beachtet wurden.

Eine Besonderheit stellen die indirekten Interaktionen dar. Beim Propagieren werden Objekte, die indirekt manipuliert worden sind, ebenfalls mit dem Kontext assoziiert. Dies geschieht zum Beispiel bei dem Umbenennen einer Klasse. Wodurch Klassen, die diese referenzieren, an den Kontext propagiert werden.

Bei der Prognose werden Objekte zum Kontext hinzugefügt, die noch nicht betrachtet wurden, aber aufgrund von Strukturabhängigkeiten wahrscheinlich bearbeitet werden.

2.2.3 Kontext Operationen

Damit verschiedene Aspekte des Entwicklungsprozesses unterstützt werden bietet Mylyn Operationen zur Manipulation von Kontexten an.

- Komposition
Es werden mehrere *Interaction Histories* zu einer kombiniert und ein neuer Kontext wird mit dem DOI Algorithmus erzeugt.

2 Related Work

- **Schneiden**
Es wird ein neuer Kontext durch schneiden des Eingangskontextes erzeugt. Dabei werden alle Ereignisse, die einem bestimmten Kriterium entsprechen, gesammelt. Dadurch lassen sich beispielsweise alle geänderten Dateien eines Kontextes ermitteln.
- **Manipulation**
Die *Interaction History* wird automatisch erstellt. Dabei kann es zu falschen Interpretationen kommen. Dem Anwender stehen dann Operationen zur Verfügung, um Objekte aus dem Kontext zu entfernen oder hinzuzufügen.
- **Induktion**
Diese Operation wird benutzt, um indirekte Ereignisse in die *Interaction History* einzufügen.

2.2.4 Architektur

Die Architektur von Mylyn folgt dem Plug-In Konzept [12] der Eclipse Plattform. Es erfolgt eine strikte Trennung zwischen der Geschäftslogik und der Benutzeroberfläche. Diese Trennung ermöglichte den Einbau und die Prüfung des Nutzens des Mylyn-Kerns im Werkzeug Protege [10].

Mylyn bietet eine Plug-In Schnittstelle zur Erweiterung und Integration mit anderen Systemen:

- **Connectors**
Diese Schnittstelle erlaubt es, Mylyn mit Werkzeugen zur Aufgabenverwaltung wie Fehler- und Anforderungsmanagement-Systemen zu verbinden.
- **VCS**
Diese Schnittstelle bietet eine Integration zu Clients von Konfigurationsmanagement-Werkzeugen. Dadurch können die Änderungen mit der Aufgabe des Entwicklers assoziiert werden.
- **Bridges**
Damit Mylyn die interne Struktur einer Datei berücksichtigen kann, muss eine Brücke zu dem jeweiligen Dateityp implementiert werden. Dadurch kann Mylyn verschiedene Dateitypen wie beispielsweise Java, C und XML berücksichtigen.
- **Monitor**
Die Monitor Schnittstelle erlaubt den Zugriff auf die Ereignisse, die der Benutzer in der IDE erzeugt.

[16]

2.2.5 Bezug zu Home Office 2.0

Die Beliebtheit von Mylyn [1] zeigt, dass der Ansatz des Aufgaben-Kontextes eine gute Unterstützung für Entwickler darstellt. Die offene und flexible Architektur bietet die Möglichkeit einer Integration mit verschiedenen Werkzeugen der Softwareentwicklung, die viele nutzen [6]. Diese Architektur wird als Vorlage für die Architektur von Home Office 2.0 genommen.

Mylyn unterstützt den Kontext innerhalb der Entwicklungsumgebung. Bei der Softwareentwicklung werden aber auch Ressourcen außerhalb der Entwicklungsumgebung genutzt, wie zum Beispiel das Web oder elektronische Kommunikationswerkzeuge. Die auf Mylyn basierende Anwendung Tasktop [7] dehnt das Konzept auf den Desktop aus.

Die Team-Unterstützung beschränkt sich auf das Übertragen von Kontexten zwischen den Entwicklern. Es gibt keine weiteren Auswertungen und Vergleiche von Kontexten, wie im Projekt ACTIVE.

Durch die verfügbaren Schnittstellen von Mylyn bietet es sich an, diese als eine Komponente kombiniert mit dem Prozessansatz von ACTIVE ins Projekt Home Office 2.0 zu integrieren.

2.3 Socio-Technical Congruence

Socio-Technical Congruence ist ein neuer Forschungsbereich des Software Engineerings. Seit dem Jahr 2008 ist dies ein eigener Workshop bei der *International Conference on Software Engineering (ICS)* [4]. Die Forschung beschäftigt sich mit dem Zusammenspiel der sozialen Struktur aller Beteiligten bei der Softwareentwicklung und den technischen Abhängigkeiten des zu entwickelnden Systems.[9]

Der Zusammenhang zwischen der Organisationsstruktur eines Teams und dem Design der Software wird als Conway's Law bezeichnet.

Melvin Conway formulierte es 1968 so:

Organizations which design software systems are constrained to produce design which are copies of the communication structures of the organizations. [20]

Socio-Technical Congruence fokussiert sich auf die Untersuchung der dynamischen Beziehung zwischen sozialen und technischen Abhängigkeiten. Dies soll zu einem besseren Verständnis der Softwareentwicklung und Optimierung des Prozesses führen. Ein besonderes Augenmerk wird dabei auf geographisch verteilte Teams gelegt. [9]

2.3.1 Socio-Technical Network

Eine Forschungsgruppe des IBM T.J. Watson Research Center hat zur Messung der Abdeckung zwischen den sozialen und technischen Abhängigkeiten ein Social-Technical Network entwickelt. Dies wird in dem Artikel *Using Software Repositories to Investigate Socio-technical Congruence in Development Projects* [23] vorgestellt.

Der Ansatz ist es, ein soziales Netzwerk-Modell mit den internen Abhängigkeiten des Softwareproduktes in einem Graphen (siehe Abb. 2.1) zu kombinieren. Dieser beinhaltet dabei drei Arten von Informationen:

- Soziales Netzwerk
In diesem Teilgraphen stellen die Beteiligten der Softwareentwicklung die Knoten dar. Die Kanten ergeben sich aus der Zusammenarbeit und Kommunikation der Personen untereinander.
- Technisches Netzwerk
Die Artefakte der Software stellen in diesem Teilgraphen die Knoten dar. Ihre Abhängigkeiten werden als Kanten modelliert.
- Arbeitsbeziehung
Die beiden Teilgraphen werden durch Arbeitsbeziehungen miteinander verbunden. Die Arbeitsbeziehung ergibt sich aus der Information welcher Mitarbeiter welches Artefakt bearbeitet.

Die Arbeitsbeziehungen werden aus Konfigurationsmanagement-Werkzeugen ermittelt. In diesen Repositories ist die Information hinterlegt welcher Mitarbeiter wann an welchem Artefakt gearbeitet hat. Das technische Netzwerk wird in einer Arbeitskopie des letzten Standes der Quellen mit statischer Codeanalyse ermittelt. Das soziale Netzwerk lässt sich aus Kollaborationswerkzeugen wie Workflow-Systeme oder Wiki's ableiten. [23]

Die Deckung zwischen dem sozialen und dem technischen Netzwerk wird durch Kantenspiegelung gemessen. Dafür wird zunächst ein Softwareartefakt ausgewählt. Über die Arbeitsbeziehung wird die Verbindung zu dem sozialen Netzwerk aufgelöst. Wenn mehrere Personen an einer Datei gearbeitet haben, wird zwischen ihnen eine Verbindung im sozialen Netzwerk erwartet. Ist diese nicht gegeben sinkt der Deckungsgrad.

Ausgehend von einem Artefakt werden dann seine Abhängigkeiten zu anderen untersucht. Die Person, die das abhängige Artefakt bearbeitet, sollte eine Verbindung im sozialen Netzwerk zu den Entwicklern des ersten Artefakts besitzen. Ist dies nicht der Fall, sinkt auch hier der Deckungsgrad.

Der so ermittelte Deckungsgrad kann ein Anzeichen für Schwächen in der Projektorganisation sein. In weiteren Arbeiten soll dies mit Qualitätsmetriken überprüft werden.

[23]

2 Related Work

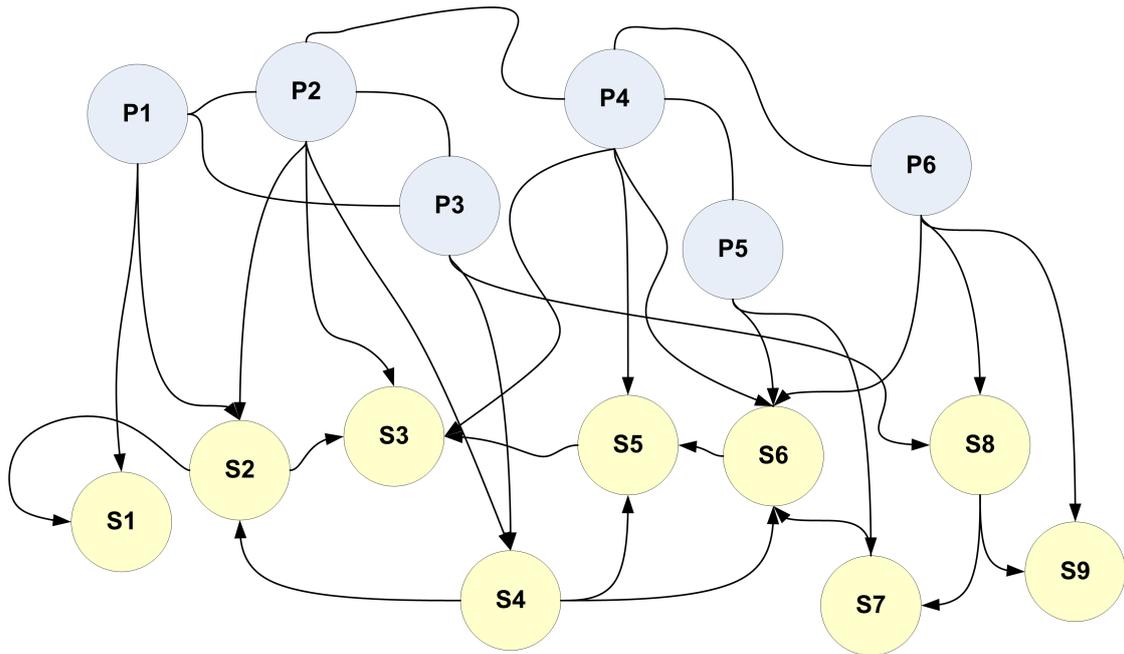


Abbildung 2.1: Schematische Darstellung eines *Socio-technical Congurrence Network*

2.3.2 Bezug zu Home Office 2.0

In verteilten Teams, wie sie im Szenario von Home Office 2.0 beschrieben sind, sind Hinweise zur Verbesserung der Kommunikation essentiell. Der Ansatz bestehende Arbeitsergebnisse zu untersuchen und mit sozialen Strukturen in Verbindung zu setzen, stellt eine weitere Informationsquelle für Home Office 2.0 dar.

Dies ermöglicht einen breiteren Blick auf den Diskurs als Kontexte ihn bieten. Die Arbeitsbeziehungen können bei der Identifikation von Experten helfen. Im Zusammenspiel mit dem Kontext einer Aufgabe, kann einem Entwickler neben den relevanten Informationen auch ein Experte angeboten werden.

3 Zusammenfassung

Die vorgestellten Arbeiten betrachten verschiedene Aspekte der Wissensarbeit und wie diese in verteilten Teams funktionieren können.

Das Projekt ACTIVE zeigt den informellen Prozess der Wissensarbeit auf und bindet ihn in den Geschäftsprozess ein. Es wird explizit die Arbeit im Team unterstützt und die Informationen eines Benutzers zur Unterstützung von Anderen zugänglich gemacht. Dies kann in verteilten Teams ein besseres Verständnis der einzelnen Mitarbeiter für ihre räumlich entfernten Kollegen geben.

Das Mylyn Projekt macht die Erzeugung des Kontextes expliziter, da die Aufgabe vor der Kontexterzeugung aktiviert werden muss. Es handelt sich hierbei um eine für die Softwareentwicklung spezialisierte Lösung, welche sich gut in den Entwicklungsprozess einbinden lässt. Da es sich im Gegensatz zu dem ACTIVE Projekt um eine open source Lösung handelt, die etabliert ist und in kommerziellen Plattformen wie Jazz [5] eingebunden ist, kann man sich ein Bild von dem Nutzen dieser Lösung machen. Die offene und flexible Architektur lässt sich auf andere Projekte übertragen und bildet das Architektur Muster für Home Office 2.0.

Diese Arbeiten betrachten den Kontext des Arbeiters bei der Erledigung einer Aufgabe. Dem gegenüber betrachtet Socio-Technical Congruence das soziale Netzwerk im Vergleich zu dem zu entwickelnden Produkt. Dadurch wird der Fokus auf das gesamte Projekt und seine Dynamik gelegt.

Der Aufgaben-Kontext, der Wissensprozess und die Betrachtung von Socio-Technical Congruence bilden Informationsquellen, die in Home Office 2.0 kombiniert werden sollen. Die Kombination aus dem aktuellen Kontext des Wissensarbeiters und dem Diskurs des Projektes können eine gute Unterstützung für den einzelnen Mitarbeiter geben, aber auch Projektprobleme frühzeitig aufzeigen.

Zum einen erlangt man Kenntnis über die aktuellen Bedürfnisse des Wissensarbeiters und auf der anderen Seite die Information über die Zusammenhänge im Projekt. Daraus ergibt sich für das Projekt Home Office 2.0 die Herausforderung aus der Auswertung bestehender Werkzeuge und den Aktivitäten der Mitarbeiter einen Kontext aufzubauen. Dieser soll dann in den Arbeitsprozess des Teams eingebettet werden.

Literaturverzeichnis

- [1] : *Eclipse Marketplace*. – URL <http://marketplace.eclipse.org/>. – Zugriffsdatum: 17. August 2010
- [2] : *Eclipse Mylyn*. – URL <http://www.eclipse.org/mylyn>. – Zugriffsdatum: 14. Juli 2010
- [3] : *EU Project Active*. – URL <http://www.active-project.eu/>. – Zugriffsdatum: 14. Juli 2010
- [4] : *International Conference on Software Engineering*. – URL <http://www.icse-conferences.org/>. – Zugriffsdatum: 14. Juli 2010
- [5] : *Jazz*. – URL <http://jazz.net/>. – Zugriffsdatum: 22. August 2010
- [6] : *Mylyn Extensions*. – URL http://wiki.eclipse.org/index.php/Mylyn_Extensions. – Zugriffsdatum: 14. Juli 2010
- [7] : *Tasktop*. – URL <http://www.tasktop.com/>. – Zugriffsdatum: 24. Juli 2010
- [8] AGILE: *Manifesto for Agile Software Development*. 02 2010. – URL <http://agilemanifesto.org/>
- [9] CATALDO, Marcelo ; EASTERBROOK, Steve ; DAMIAN, Daniela ; HERBSLEB, James ; DEVANBU, Premkumar ; MOCKUS, Audris: 2nd international workshop on socio-technical congruence (STC 2009). (2009), S. 476–477. ISBN 978-1-4244-3495-4
- [10] D’ENTREMONT, T. ; STOREY, M.-A.: Using a Degree-of-Interest Model for Adaptive Visualizations in Protege. (2006)
- [11] DRUCKER, Peter F.: *Management. An abridged and revised version of Management: Tasks, Responsibilities, Practices*. Butterworth-Heinemann, Oxford, 1991
- [12] ERIC CLAYBERG, Dan R.: *Eclipse: Building Commercial-Quality Plug-ins*. Addison-Wesley, 2008
- [13] FONDATION, Eclipse: *Eclipse*. – URL <http://www.eclipse.org/>. – Zugriffsdatum: 14. Juli 2010
- [14] GOMEZ-PEREZ, Jose M. ; GROBELNIK, Marko ; RUIZ, Carlos ; TILLY, Marcel ; WARREN, Paul: Using task context to achieve effective information delivery. (2009), S. 1–6. ISBN 978-1-60558-528-4

Literaturverzeichnis

- [15] HUNT, Andy: *Pragmatic Thinking and Learning: Refactor Your Wetware (Pragmatic Programmers)*. Pragmatic Bookshelf, 2008. – ISBN 1934356050, 9781934356050
- [16] KERSTEN, Mik: *Focusing knowledge work with task context*. Vancouver, BC, Canada, Canada, Dissertation, 2007
- [17] KIRSH, David: A Few Thoughts on Cognitive Overload. In: *Intellectica* 30 (2000)
- [18] MCAFEE, Andrew P.: Enterprise 2.0: The Dawn of Emergent Collaboration. In: *MITSloan Management Review* 47 (2006), Nr. 3, S. 21–28. – URL <http://sloanreview.mit.edu/the-magazine/articles/2006/spring/47306/enterprise-the-dawn-of-emergent-collaboration/>
- [19] PANIER, Karsten: *Home Office 2.0 - Collaborate Working*. 2010. – URL <http://users.informatik.haw-hamburg.de/~ubicomp/projekte/master09-10-aw1/Panier/bericht.pdf>. – Zugriffsdatum: 14. Juli 2010
- [20] RANDELL, B.: Software engineering in 1968. (1979), S. 1–10. ISBN none
- [21] SCHILIT, Bill ; ADAMS, Norman ; WANT, Roy: *Context-Aware Computing Applications*. 1994
- [22] THOMAS H. DAVENPORT, M.C. B.: Improving Knowledge Work Processes. In: *Sloan Management Review* (1996)
- [23] VALETTO, Giuseppe ; HELANDER, Mary ; EHRLICH, Kate ; CHULANI, Sunita ; WEGMAN, Mark ; WILLIAMS, Clay: Using Software Repositories to Investigate Socio-technical Congruence in Development Projects. (2007), S. 25. ISBN 0-7695-2950-X