

Formularbasierte Traceability-Unterstützung (Positionspapier)

Markus Ueberall

Professur ABVS, Institut für Informatik, Johann-Wolfgang-Goethe-Universität,
D-60054 Frankfurt/Main, ueberall@tm.informatik.uni-frankfurt.de

1 Motivation

Unzureichende Kommunikation zwischen den prozeßbeteiligten Personen ist ein zentrales Problem jedes Entwicklungsprozesses; insbesondere drohen hierbei Informationsverluste aus nicht-zeitnahe Abgleichen von Anforderungen und deren Umsetzungen. Dieser Umstand wird in der Praxis heutzutage dadurch verschärft, daß Entwicklungszyklen zunehmend beschleunigt und Projekte entweder vollständig ausgelagert oder gemeinsam mit wechselnden externen Dienstleistern umgesetzt werden, wobei aus Planungsgründen häufig die Dokumentation vernachlässigt wird. Vorschnele Abnahme von Dokumentation oder unzureichende Spezifikation gewünschter Funktionalitäten führen mittel- und langfristig zu Problemen und damit zu u.U. extrem hohen Kosten. Gute Lösungen für die Traceability-Unterstützung [S02], welche zum Abbau des Dokumentationsdefizits führen, bringen daher eine erhebliche Kostenreduktion mit sich.

Das Problem bei zeitnahe, *prozeßbegleitender* Dokumentation ist der damit verbundene Aufwand, gerade bei Entwicklung großer softwareintensiver Systeme, welche nicht selten auf parametrisierbare Fremdsysteme zugreifen (müssen). In der Regel gibt es zudem konzernweit vorgeschriebene Entwicklungswerkzeuge für bestimmte Aufgaben (Quellcode-Verwaltung, Change/Defect Management, ...), welche untereinander keinen Informationsaustausch ermöglichen. Gefordert ist daher eine leichtgewichtige, schnelle und möglichst einfache Sammlung von Informationen, wobei es nicht darauf ankommt, bereits in Verwendung befindliche Programme/Dokumente zu *ersetzen* (aufgrund von Konzernrichtlinien wäre das in vielen Fällen auch gar nicht möglich), sondern sie zu *ergänzen* und konsistente Querverweise erstellen bzw. pflegen zu können.

Im Rahmen der laufenden Dissertation soll auf Basis einer Analyse aktueller Entwicklungsabläufe und verwendeter Werkzeuge ein Lösungsvorschlag konzipiert und prototypisch implementiert werden, welcher alle Phasen eines Entwicklungsprozesses abdecken kann.

2 Methodische Basis des Ansatzes

Kern des Ansatzes ist ein alle Phasen eines Entwicklungsprozesses umfassendes Gesamtmodell, welches die Konsistenz der Artefakte aller Phasen gewährleistet. Dieses Gesamtmodell [UD05] besteht aus einem generischen Metaprozeßmodell und der Verwendung von vorge-

gebenen, jedoch erweiterbaren terminologiebasierten Ontologien [SS06]. Mithilfe der Ontologien können zunächst sehr frei formulierte Aussagen sukzessiv auf formale Konzepte abgebildet werden. Sowohl die Abbildung von Konzepten innerhalb derselben Domäne als auch die *erzwungene* Verbindung von Konzepten unterschiedlicher Domänen/Phasen wird durch das Metaprozeßmodell gelenkt, welches die sofortige bzw. verzögerte Validierung und die automatische Protokollierung aller Benutzereingaben steuert.

Vorgenannte Ausgangsidee erlaubt die Nachbildung bekannter, auch komplexer Vorgehensmodelle; es ist jedoch wichtig, daß zunächst anhand einer einfachen Fallstudie eine Konzentration auf einzelne Prozeßschritte und wenige domänenspezifische Konzepte stattfindet, um die initiale Anforderungsanalyse des zu erstellenden Prototypen zu erleichtern. Parallel zur Betrachtung möglicher Erweiterungen in Form von zusätzlichen Prozeßschritten und einzubindenden Werkzeugen (bspw. Entwicklungsumgebungen, Versionsverwaltungssysteme), welche ihrerseits einer Evaluation zu unterziehen sind, läßt sich sowohl die Modellierung überarbeiten als auch eine prototypische Umsetzung – möglichst unter Verwendung ebenjener Werkzeuge – realisieren. Diese in der Softwaretechnik mögliche vorteilhafte Kombination von Einsatz und Evaluation von Werkzeugen tröstet ein wenig über die in der Regel sehr arbeitsintensive Realisierung hinweg.

Zur Unterstützung der Anwender kommen im vorliegenden Fall sogenannte *Schablonen* für ontologiebasierte Artefakte zum Einsatz [UD06]. Diese dienen nicht nur zur Angabe der Attribute von ihnen referenzierter Konzept(instanz)en und deren Beziehungen untereinander, sondern sie spezifizieren auch deren Verwendungsmöglichkeiten; Queries dienen hierbei zur Filterung relevanter Informationen, Constraints erlauben u.a. die Formulierung von Abhängigkeiten zwischen Artefakten.

Durch die erzwungene Verknüpfung von Artefakten verschiedener Entwicklungsphasen ist gewährleistet, daß jeder Prozessbeteiligte Zugriff auf relevante Informationen erhält, welche durch andere beigesteuert werden. Auf diese Weise läßt sich in der Entwurfs- und der Implementationsphase auch die Adressierung spezifizierter nichtfunktionaler Anforderungen erzwingen, da diese dann etwa in Architekturmodellen und Klassen explizit referenziert werden müssen – wenngleich die eigentliche Validierung der getroffenen Aussagen in diesem Fall nach wie vor manuell erfolgen muß.

Da sich die vorgenannten Schablonen relativ einfach in Eingabemasken umsetzen lassen, läßt sich somit relativ schnell eine einfache Benutzerschnittstelle entwerfen, um einzelne Aktionen aus der Fallstudie mit einem ersten Prototypen praktisch nachzuvollziehen. Im vorliegenden Fall bot sich eine Realisierung als Eclipse Plug-in an, um einerseits auf eine Vielzahl von vordefinierten Funktionen und Schnittstellen dieser frei verfügbaren Entwicklungsumgebung zurückgreifen zu können und andererseits aufgrund ihrer Verbreitung eine bessere Ausgangsbasis für die erforderliche Fremdevaluation des Prototypen zu schaffen.

Alle auf terminologieorientierten Ontologien basierenden Informationen werden intern durch Topic Maps [ISO1] repräsentiert, deren themenzentrische Konzeption bereits der Sichtweise des Entwicklungsprozesses entspricht. Auch die geforderte Freiheit der Organisation von Konzepten und Beziehungen ist gewährleistet, da Topics und Assoziationen innerhalb der betrachteten spezifischen topic map keinerlei Beschränkungen unterliegen und beliebig annotiert werden können.

Analog zum Einsatz von Java/Eclipse stand auch bei der Wahl des Repräsentationsformats die Verfügbarkeit zahlreicher existierender Hilfswerkzeuge und die einfache Einsatzmöglichkeit im Vordergrund, um die obengenannten Aufwände für die praktische Umsetzung zu begrenzen.

Literatur

- [ISO1] ISO/IEC IS 13250-2: 2006: Information Technology – Document Description and Processing Languages – Topic Maps – Data Model. International Organization for Standardization, Geneva, Switzerland. <http://www.isotopicmaps.org/sam/>
- [SS06] Schmitz-Esser, W., Sigel, A.: Introducing Terminology-based Ontologies. Proc. 9th International Conference of the International Society for Knowledge Organization (ISKO), 2006. <http://eprints.rclis.org/archive/00006612/>
- [S02] Strašunskas, D.: Traceability in Collaborative Systems Development from Lifecycle Perspective. Position Paper, Proceedings of the 1st International Workshop on Traceability, 2002. <http://www.idi.ntnu.no/~dstrasun/papers/strasunskas2002.pdf>
- [UD05] Ueberall, M., Drobnik, O.: Collaborative Software Development and Topic Maps. Proc. 1st International Workshop on Topic Maps Research and Applications (TMRA 2005). Springer Lecture Notes in Artificial Intelligence (LNAI) 3873 169–176. http://dx.doi.org/10.1007/11676904_15
- [UD06] Ueberall M., Drobnik, O.: On Topic Map Templates and Traceability. Proc. 2nd International Workshop on Topic Maps Research and Applications (TMRA 2006). To appear in: Springer Lecture Notes in Artificial Intelligence (LNAI)

Angaben zur Person

Markus Ueberall studierte ab 1991 (mit Unterbrechung) Informatik an der JWGU Frankfurt/Main; im Rahmen des selbstgewählten Diplomarbeitsthemas spezialisierte er sich hierbei insbesondere auf verteilte Anwendungen und Information Retrieval.

Nach Abschluß des Studiums (12/2001) war er im Rahmen des durch das DFG-SPP1140 geförderten Projekts ELAN (E-Learning in mobile Ad-hoc Networks) zwei Jahre als wissenschaftlicher Mitarbeiter am Lehrstuhl ABVS/Telematik tätig; eigener Schwerpunkt hierbei war die Untersuchung der Anforderungen modularer, konstruktivistischer E-Learning-Applikationen in P2P-Netzen, insbesondere die Modellierung sogenannter „Lerneinheiten“ (auf Topic Maps-Basis) und der Protokollierung ihrer Nutzung.

Seit 08/2004 arbeitet er als IT-Berater auf dem Finanzdienstleistungssektor; dies ermöglicht praktische Einblicke in verschiedene komplexe Workflows, was mit zur Übertragung der bisherigen Ansätze und Erkenntnisse (die Modellierung und Verfolgung von Informationsobjekten betreffend) in den vorgenannten Kontext anregte.