

Unterschiedliche Präsentationen von Aufgabenmodellen

Daniel Reichart, Anke Dittmar, Peter Forbrig, Maik Wurdel

Universität Rostock

Zusammenfassung

Aufgabenmodelle können auf unterschiedlichste Art und Weise Visualisiert werden. Das Papier fokussiert auf die Möglichkeit der Werkzeugunterstützung für die Nutzung von Editoren die gleichzeitig die Bearbeitung eines Modells in unterschiedlichen Darstellungen ermöglicht. Die Editoren sind in Java geschriebene Plugins für die Eclipse-Entwicklungsumgebung.

1 Einleitung

Aufgabenmodelle haben sich speziell bei der Anforderungsanalyse als hilfreiches Mittel erwiesen. Dabei hat sich in der HCI-Community mit (CTT 2008) herausgestellt, der aber außerhalb dieser Community mitunter auf Akzeptanzprobleme stößt. Ein Grund könnte dafür sein, dass Aufgabenmodelle nicht zum Standard von (UML 2008) gehören. In (Brüning et. al. 2007) haben wir bereits Beziehungen zwischen Aktivitätsdiagrammen und Aufgabenmodellen diskutiert. Nachfolgend werden wir kurz vorstellen, wie Aufgabenmodelle als Aktivitätsdiagramme präsentiert werden können. Dabei wird vorausgesetzt, dass der Leser etwas mit der Notation von Aufgabenmodellen vertraut ist.

2 Repräsentationen von Aufgabenmodellen

Wir haben schon erwähnt, dass sich mit CTT ein gewisser Standard für Aufgabenmodelle herausgebildet hat. Unsere Modelle wurden ursprünglich etwas anders notiert, wobei wir Rollen, Arbeitsgegenstände, Werkzeuge und Geräte zu den Aufgaben zuordnen. Das kann aber auch in einen CTT-artigen Stil geschehen. Zu diesem Zwecke wurde ein eigenes (Eclipse 2008) – Plugin implementiert, dessen Visualisierung in Abbildung 1 zu sehen ist.

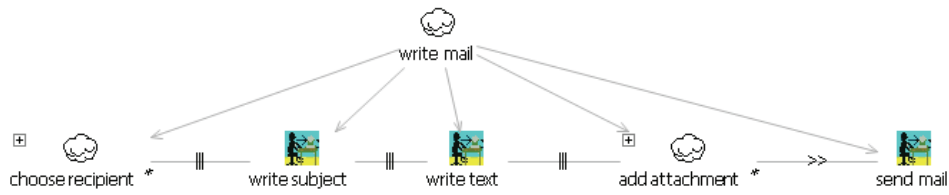


Abbildung 1: Aufgabenmodell zum Schreiben einer Mail in einer CTTE-artigen Notation

Mit einer anderen Darstellung kann das vermieden werden. In Abbildung 2 ist gleich klar, dass was zu nun ist, bevor “send mail” ausgeführt werden kann. Diese Darstellung wird durch ein weiteres Plugin realisiert.

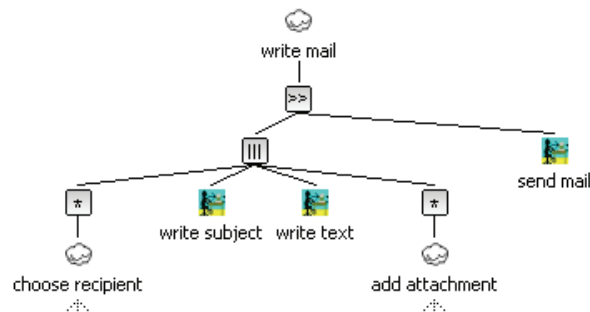


Abbildung 2: Operatorzentrierte Repräsentation eines Aufgabenmodells

Es wurde bereits erwähnt, dass die dargestellten Aufgabenmodelle nicht Teil der Sprache UML sind. Daher scheint es sinnvoll, den Softwareentwicklern Aufgabenmodelle als Aktivitätsdiagramme zu präsentieren. Auch dafür wurde ein eigenes Plugin entwickelt.

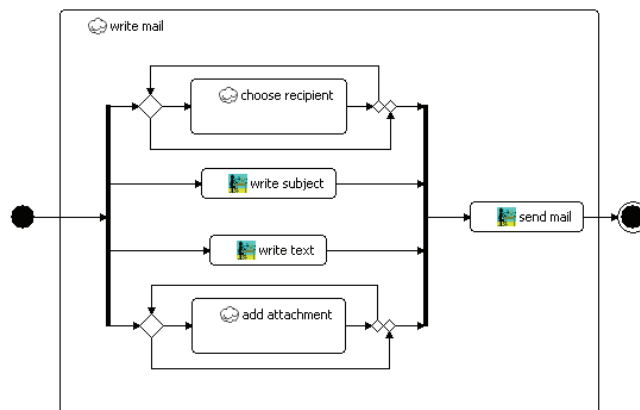


Abbildung 3: Aufgabenmodell zum Schreiben einer Mail als Aktivitätsdiagramm

Aufbauend auf Aufgabenmodellen, die schon animiert werden können, erfolgt nach unserer Vorgehensweise die Spezifikation von Navigationsdialogen (Wolff et. al 2006). Abbildung 4 gibt einen Eindruck, wie ein hierarchischer Dialoggraph aussieht. Während der Demonstration wird gezeigt werden, wie animierte abstrakte kanonische Prototypen und animierte konkrete Benutzungsoberflächen für unterschiedliche Geräte erzeugt werden können.

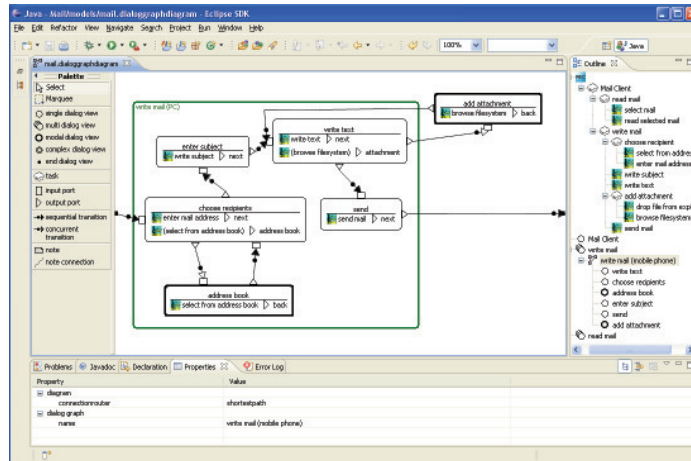


Abbildung 4: Hierarchische Spezifikation eines Navigationsmodells

3 Modellbasierte Werkzeugunterstützung

Basierend auf Metamodellen können mit Hilfe von (Eclipse 2008) Editoren generiert werden, mit denen man Aufgabenmodelle bearbeiten kann. Diese generierten Editoren sind nicht sehr benutzungsfreundlich. Durch (EMF 2008) ist man aber in der Lage, gut handhabbare Editoren zu erstellen, die auf den gleichen Daten arbeiten wie die generierten.

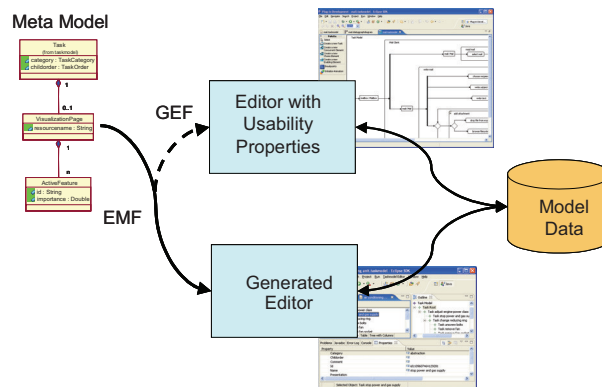


Abbildung 5: Modelle, Daten und Editoren

Abbildung 4 gibt einen Eindruck vom Entwicklungsprozess bei der Nutzung des EMF- und GEF-Frameworks.

4 Zusammenfassung und Ausblick

Es wurde gezeigt, wie die gleichen Daten eines Aufgabenmodells auf ganz unterschiedliche Art und Weise präsentiert werden können. Durch die Bereitstellung von Editoren für die unterschiedlichen Präsentationsformen sollen die Konzepte von Aufgabenmodellen attraktiver gemacht werden und breitere Benutzergruppen ansprechen.

Es kann dabei auch eine gegenseitige Befruchtung der Konzepte geben. Temporale Beziehungen aus Aufgabenmodellen, die nicht in UML-Aktivitätsdiagrammen enthalten sind, können dort als Stereotypen dargestellt werden. Eine Animation kann sofort auf der Basis der klassischen Aufgabenmodelle erfolgen.

Auf der Basis der Eclipse-Frameworks wurden Werkzeuge entwickelt, die von der Analyse über den Entwurf von Navigationsspezifikationen bis zur Animation einer konkreten Benutzungsoberfläche reichen. Der Übergang von der Spezifikation der Aufgabenmodelle zu Navigationsdialogen konnte im Papier nur ganz kurz angeschnitten werden, wird bei der Demonstration aber durch Anwendungsbeispiele untermauert werden. Die Demonstration kann auch zeigen, wie bereits entwickelte Modelle in Form von Patterns und Komponenten in den Entwicklungsprozess einbezogen werden können.

Literaturverzeichnis

- Berti, S., Correani, F., Mori, G., Paternò, F. & Santoro, C. (2004). A transformation-based environment for designing multi-device interactive applications. In: *Proceedings of the 9th international conference on Intelligent user interfaces*. Funchal, Januar 2004.
- Brüning, J., Dittmar, A., Forbrig, P., Reichart, D.(2007). Getting SW Engineers on Board: Task Modelling with Activity Diagrams, In: *Proc. of EIS 2007*, Salamanca, Spain, March 2007.
- CTTE (2008). <http://giove.cnuce.cnr.it/ctte.html> (eingesehen: 20.02.08)
- Eclipse (2008). <http://www.eclipse.org/> (eingesehen: 20.02.08)
- EMF (2008). <http://www.eclipse.org/emf/> (eingesehen: 20.02.08)
- GEF (2008). <http://www.eclipse.org/gef/> (eingesehen: 20.02.08)
- MDA (2008). <http://www.omg.org/mda/> (eingesehen: 20.02.08)
- Mori, G., Paternò, F. & Santoro, C (2004). Design and Development of Multidevice User Interfaces through MultipleLogical Descriptions, *IEEE Transactions on Software Engineering*, August 2004, S.507-520.
- UML (2008) <http://www.uml.org> (eingesehen: 20.02.08)
- Wolff, A., Forbrig, P., Dittmar, A. & Reichart, D. (2006). Tool Support for an Evolutionary Design Process using Patterns, Proc. of the Workshop: Multi-channel Adaptive Context-sensitive (MAC) Systems: Building Links Between Research Communities, Glasgow, 2006.