

Ein Werkzeug zur Spezifikation von Dialoggraphen

Peter Forbrig, Daniel Reichart

Universität Rostock, Institut für Informatik, Lehrstuhl für Softwaretechnik

Zusammenfassung

Die Anpassung der Navigation einer Anwendung an die Bedingungen unterschiedlicher Geräte ist einerseits eine wichtige Aufgabenstellung für die Wartung und andererseits bedeutend für die Benutzbarkeit von Softwaresystemen. Der Beitrag präsentiert eine Vorgehensweise, wie auf der Basis von Aufgabenmodellen hierarchische Dialogspezifikationen entwickelt werden können.

1 Einleitung

Der Prozess der Softwareentwicklung kann als eine Folge von Transformationen angesehen werden. Verschiedene Ausgangsmodelle werden in andere Modelle durch automatische Transformation, Anwendung von Patterns oder Implementation überführt. Die Nutzung gemeinsamer Ausgangsmodelle für die Entwicklung des Applikationskerns und den Entwurf der Benutzungsoberfläche soll die Kluft zwischen Softwareentwicklern und UI-Designern überwinden und eine gemeinsame Nutzung von Analysedokumenten fördern.

1.1 Aufgabenmodell

Wie bei vielen modellbasierten Entwicklungsansätzen stehen auch bei unserem Ansatz Aufgabenmodelle im Mittelpunkt. Sie sind jedoch nicht isoliert, sondern haben Relationen zu Objekten und Benutzern. Gleichzeitig stehen Aufgaben in Beziehung zur Umgebung.

Aufgabe = (Ziel, Unteraufgaben, Temporale Beziehungen, Vorbedingung, Nachbedingung, Arbeitsgegenstand, Werkzeuge, Rollen, Geräte)

Das Umgebungsmodell besteht dabei stark vereinfacht nur aus der Angabe des benötigten Gerätes. Die Geräte sind wiederum durch eine „>“-Beziehung charakterisiert, die vereinfacht spezifiziert, dass ein Gerät mehr Fähigkeiten besitzt als ein anderes.

1.2 Dialogmodell

Für die Erzeugung von Dialogmodellen existieren unterschiedliche Ideen. Dialoge können sich direkt aus dem Aufgabenmodell ergeben wie im TERESA-Projekt (Berti et al 2004). Wir verfolgen den Ansatz der expliziten Navigationspezifikation mittels abstrakter Dialoggraphen (Schlungbaum & Elwert 1996) mit Verknüpfungen zu Aufgaben (Forbrig et al 2006). Dialoggraphen fassen dabei Funktionalitäten in Dialogsichten zusammen. Mögliche Übergänge zwischen Dialogsichten werden mittels Transitionen spezifiziert und können durch die Aktivierung von Aufgaben ausgelöst werden.

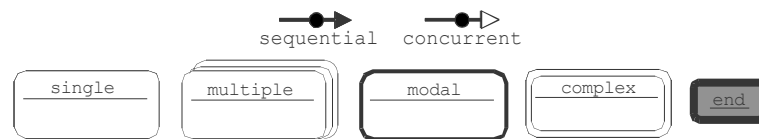


Abbildung 1: Symbole für Übergänge und Dialogsichten

Wünschenswert ist ein gemeinsames plattformübergreifendes Ausgangsmodell, welches anschließend an andere Geräteklassen angepasst wird. Auf einem Notebook können natürlich mehr Informationen gleichzeitig angezeigt werden als auf einem Handy. Es müssen also entweder Notebook-Dialogsichten für das Handy weiter unterteilt werden oder ausgehend vom Dialogmodell fürs Handy mehrere Dialoge zu einem Dialog zusammengefasst werden.

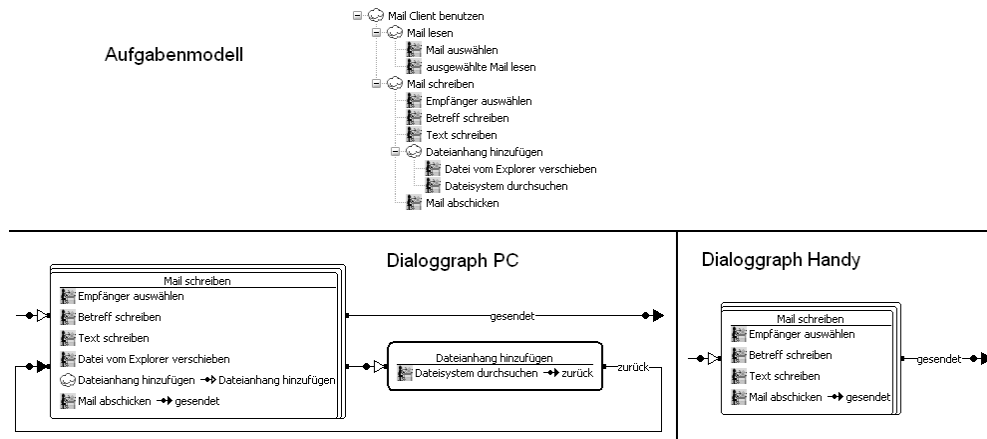


Abbildung 2: Unterschiedliche Dialoggraphen für 2 Zielplattformen

Bei der Transformation eines abstrakten Dialoggraphen in einen plattformspezifischen Dialoggraphen spielt das Aufgabenmodell eine wichtige Rolle. Teilaufgaben, die auf der Zielplattform nicht bearbeitet werden können, sind auch aus dem Zieldialoggraph zu streichen.

Abbildung 2 zeigt oben das Aufgabenmodell einer Mailapplikation und unten Ausschnitte aus Dialoggraphen für 2 Zielplattformen.

2 Werkzeugunterstützung

Zur Unterstützung des im vorigen Kapitel beschriebenen Entwicklungsansatzes wurden im Vorfeld bereits entwickelte Werkzeuge (Forbrig & Reichart 2006) erweitert. Zu diesen zählen neben Editoren für Aufgabenmodelle, Aktivitätsdiagramme und Dialoggraphen auch Werkzeuge zur Simulation dieser Modelle und zur Generierung von Quellcode für verschiedene Zielplattformen. Für jedes der verwendeten Modelle ist ein Metamodell in UML spezifiziert. Dieses dient nicht nur der formalen Spezifikation der im Entwicklungsprozess angewandten Modelle. Es lassen sich daraus mit Hilfe des Eclipse Modeling Framework (EMF) komplette Bearbeitungswerkzeuge generieren, welche die Grundlage unserer gesamten Werkzeugentwicklung darstellen. Jedes der erzeugten Werkzeuge wurde als Plug-In in die Entwicklungsumgebung Eclipse integriert.

Zur Integration der Modularisierungskonzepte in den Dialoggrapheditor wurde unter anderem das Metamodell für Dialoggraphen um Portinformationen erweitert. Eine ausführliche Spezifikation des Metamodells ist in (Rieckhoff 2006) zu finden.

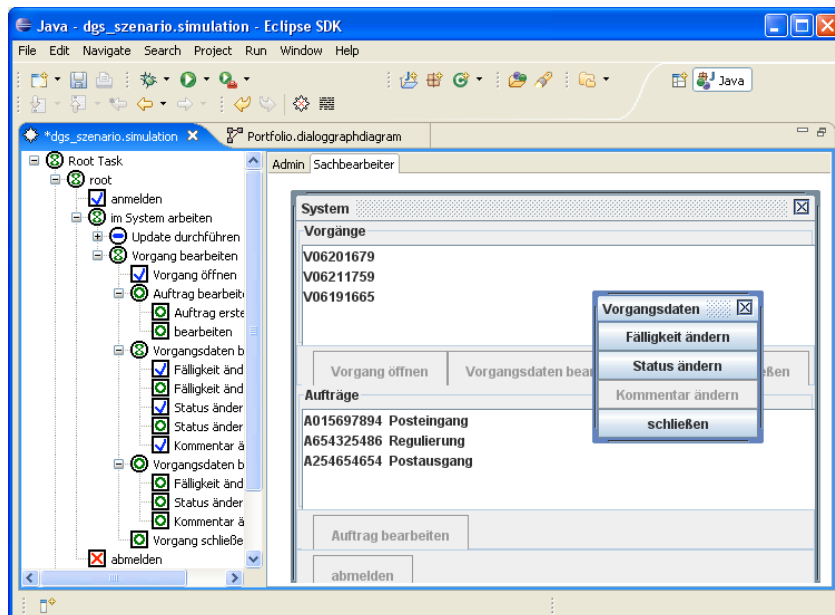


Abbildung 3: Simulation eines Dialoggraphen

Neben den Änderungen an den Metamodellen wurde der ursprüngliche Editor um einige Funktionalitäten erweitert. Diese umfassen die Generierung geräteabhängiger Dialoggraphen, das Definieren und Entfernen von Ports und die Erzeugung von Unterdialoggraphen für beliebige Dialogsichten. Abbildung 3 zeigt einen Screenshot des erweiterten Dialoggrapheditors.

3 Zusammenfassung und Ausblick

Wir haben in diesem Beitrag gezeigt, welche Bedeutung die Anpassung von Dialogmodellen an verschiedene Geräte hat und einige Probleme und Lösungsmöglichkeiten aufgezeigt. Die Modularisierung von Teilen des Dialogmodells erlaubt Kapselung, Austausch und Wiederverwendung häufig benötigter Spezifikationen. Es wurde auch ein Verfahren gezeigt, mit dem abstrakte Dialoggraphen an einzelne Zielplattformen angepasst werden können.

Aufgrund der stetig wachsenden Gerätevielfalt wird in Zukunft eine Weiterentwicklung der gezeigten Techniken erforderlich sein. Hier spielt die Identifikation und Anwendung von Patterns eine wichtige Rolle. Eine weitere Aufgabe für die Zukunft ist die Anwendung von Usability-Testing-Methoden, um eine hohe Qualität der modellbasiert erzeugten Nutzungsschnittstellen sicherzustellen. Aufgrund steigender Leistungsfähigkeit mobiler Geräte ist auch die Einbeziehung anderer Interaktionstechniken wie z.B. Sprachinteraktion in den Modellierungsprozess sinnvoll.

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.
- Forbrig, P.; Fuchs, G.; Reichart, D.; Schumann, H. (2006): Modellbasierte Entwicklung mobiler multimodaler Nutzungsschnittstellen. Mensch und Computer. Gelsenkirchen, September 2006.
- Forbrig, P.; Reichart, D. (2006): Modellbasierte Entwicklung modellbasierter Werkzeuge. 1. Workshop der Special Interest Group „Model-Driven Software Engineering“ der gasq. Hamburg, Dezember 2006.
- Rieckhoff, S. (2006): Modularisierung von Navigationsspezifikationen unter Beachtung von HCI-Patterns, Diplomarbeit, Universität Rostock, 2006.
- Schlungbaum, E.; Elwert, T. (1996): Dialogue Graphs – A Formal and Visual Specification Technique for Dialogue Modelling. Springer, 1996.

Kontaktinformation

Universität Rostock
Fakultät für Informatik und Elektrotechnik
Albert-Einstein-Str. 21
18051 Rostock
[peter.forbrig | daniel.reichart]@informatik.uni-rostock.de