

Spezifikation von „Multiple User Interfaces“ mit Dialographen

Peter Forbrig, Daniel Reichart

Universität Rostock
Institut für Informatik, Lehrstuhl Softwaretechnik
Albert-Einstein-Str. 21
18051 Rostock
[peter.forbrig|daniel.reichart]@informatik.uni-rostock.de

Abstract: Die Anpassung der Navigation einer Anwendung an die Bedingungen unterschiedlicher Geräte ist einerseits eine wichtige Aufgabenstellung für die Softwaretechnik. Der Beitrag präsentiert eine Möglichkeit, wie eine automatische Anpassung an unterschiedliche Geräte erfolgen kann.

1 Einleitung

Die zunehmende Verschmelzung von Internet, mobiler Telephonie und tragbaren Computern führt zu völlig neuen interaktiven Systemen. Für die Entwicklung derartige Systeme ist ein Entwicklungsprozess basierend auf Modellinformationen sehr hilfreich. Abbildung 1 fasst unsere Vorstellungen vom Softwareentwicklungsprozess kompakt zusammen.

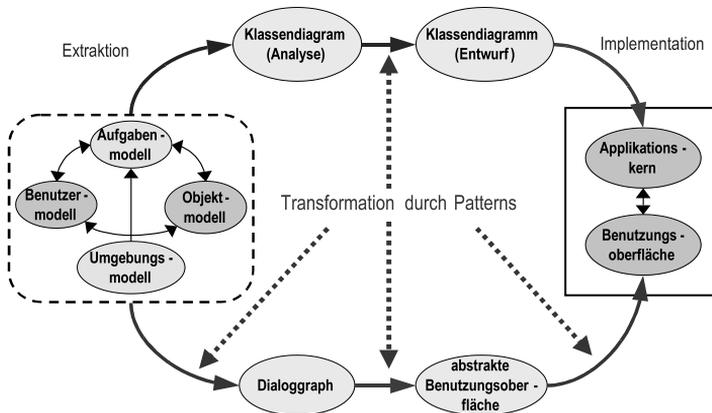


Abbildung 1: Modell der Softwareentwicklung

Die Nutzung gemeinsamer Ausgangsmodelle für die Entwicklung des Applikationskerns und den Entwurf der Benutzungsoberfläche ist ganz entscheidend und soll die Kluft zwischen Softwareentwicklern und UI-Designern überwinden und eine gemeinsame Nutzung von Analysedokumenten fördern.

2 Modellbasierter Entwicklungsansatz

2.1 Aufgabenmodell

Wie bei vielen modellbasierten Entwicklungsansätzen in der HCI-Community stehen auch bei unserem Ansatz Aufgabenmodelle im Mittelpunkt. Sie stehen jedoch nicht allein, sondern haben Beziehungen zu Objekten und Benutzern aus dem Anwendungsbereich. Gleichzeitig stehen Aufgaben in Beziehung zur Umgebung. So ist es nicht immer sinnvoll, alle Aufgaben auf allen Geräten auszuführen. So wird beispielsweise kaum jemand Probekapitel zu einem Buch auf seinem Handy durchlesen. Für diese Aufgabe sollte mindestens ein Laptop zur Verfügung stehen.

In diesem Sinne sei eine Aufgabe zunächst als 9-Tupel beschrieben.

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

2.2 Dialogmodell

Dialoge können sich direkt aus dem Aufgabenmodell bzw. aus den theoretisch möglichen Zuständen des Aufgabenmodells ergeben wie im TERESA-Projekt [BC04]. Dabei wird von der These ausgegangen, dass alle zur gleichen Zeit aktivierbaren Aufgaben auch zur gleichen Zeit sichtbar gemacht werden. Das ist für kleine Aufgabenmodelle noch hinnehmbar, für umfangreichere Modelle ergibt sich aber leicht eine Überfrachtung der Benutzungsoberflächen. Dem Nutzer muss ja nicht unbedingt zu jeder Zeit alles präsentiert werden, was im aktuellen Zustand des Systems möglich ist.

Wir verfolgen daher den Ansatz des expliziten Entwurfs einer Navigationsspezifikation. Das geschieht mittels abstrakter Dialoggraphen [SE96] und der Verknüpfung der Knoten der Dialogspezifikation mit Aufgaben aus dem Aufgabenmodell [FR06]. Dialoggraphen fassen Aufgaben in Dialogsichten zusammen. Mit Hilfe der Dialoggraphen können plattformspezifische Navigationen beschreiben werden. Das ist der wesentliche Unterschied zu [LH00], [QT06] und vielen anderen Ansätzen, die sich auf die Modellierung von allgemeinen Web-Anwendungen konzentrieren. Gerätespezifische Spezifikationen werden dabei nicht genutzt. Auch Überlegungen zur Modularisierung werden dabei nicht angestellt. Abbildung 3 präsentiert die verschiedenen Symbole die für die Übergänge und die Knoten eines Graphen möglich sind.

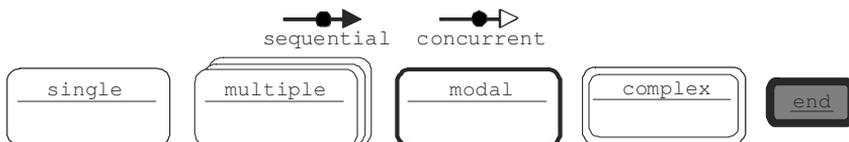


Abbildung 2: Symbole für Übergänge und Dialogsichten

Einfache Sichten werden als „single“ charakterisiert. Gibt es mehrere Instanzen einer Sicht. So muss der entsprechend Knoten als „multiple“ spezifiziert sein. Ist ein Knoten als „modal“ charakterisiert, so gestattet er bei der Abarbeitung nicht die gleichzeitige Bearbeitung einer nebenläufig (concurrent) präsentierten Sicht. Knoten, die als „complex“ charakterisiert wird beinhalten wiederum einen Dialoggraphen.

Geräteübergreifende Dialogspezifikation: Die Nutzung eines Dialoggraphen für mehrere Zielplattformen bringt einige Probleme mit sich. Unterschiedliche Displaygrößen erfordern eine unterschiedliche Granularität der Dialogspezifikation. Auf einem Notebook können natürlich mehr Informationen gleichzeitig angezeigt werden als auf einem Handy. In Abbildung 4 werden beispielsweise die Dialoge 1 bis 3, die für ein mobiles Gerät in dieser Weise spezifiziert wurden, zu einem einzelnen Dialog zusammengefasst.

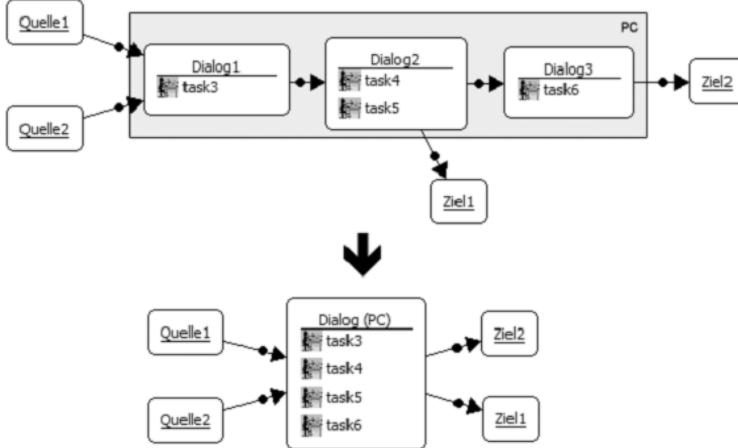


Abbildung 3: Zusammenfassung mehrerer Dialogsichten

Bei der Transformation eines abstrakten Dialoggraphen in einen plattformspezifischen Dialoggraphen spielt auch das Aufgabenmodell eine große Rolle. Teilaufgaben, die auf der Zielplattform nicht bearbeitet werden können, sind auch aus dem Zieldialoggraph zu streichen. Dabei kommt es vor, dass überflüssige Dialogsichten entstehen. Abbildung 3 zeigt oben das Aufgabenmodell einer Mailapplikation und unten Ausschnitte aus Dialoggraphen für 2 Zielplattformen. Es fällt auf, dass einige Aufgaben, die auf dem Handy nicht verfügbar sind, wegfallen. Die Dialogsicht Dateianhang hinzufügen entfällt vollständig, da alle Aufgaben, die in dieser Dialogsicht enthalten sind, auf diesem Gerät nicht bearbeitet werden können.

Bei einer Transformation eines abstrakten plattformunabhängigen Dialoggraphen in einen konkreten Dialoggraphen für eine Zielplattform sind folgende Schritte erforderlich: (1) Zusammenfassung mehrerer Dialogsichten, (2) Entfernung der auf dem Gerät nicht verfügbaren Aufgaben und (3) Entfernung leerer Dialogsichten und Anpassung der Navigationsstruktur

Auf den letzten Schritt soll hier noch etwas näher eingegangen werden.

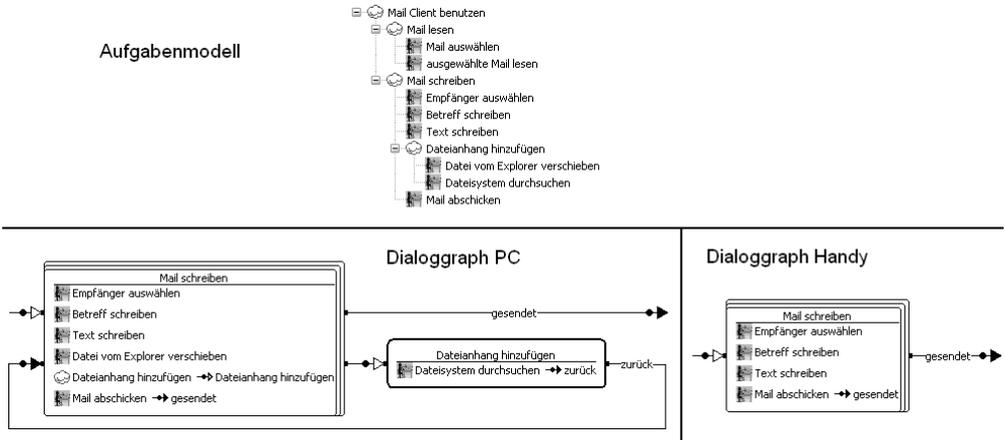


Abbildung 4: Unterschiedliche Dialoggraphen für 2 Zielplattformen

Im vorangegangenen Beispiel war die Vereinfachung des Dialoggraphen trivial, da für die zu löschende Dialogsicht nur Transitionen von und zu einer anderen Dialogsicht existierten. Im allgemeinen Fall müssen alle der zu löschenden Dialogsicht vorausgehenden mit allen nachfolgenden Dialogsichten per Transition miteinander verknüpft werden. Eingehende und ausgehende Transitionen werden so durch das Löschen der Dialogsicht vereint. Es ist offensichtlich, dass bei sequentiellm Eingang und sequentiellm Ausgang auch beim Löschen der dazwischen liegenden Sicht eine Sequenz verbleibt. In allen anderen Fällen ergibt sich Nebenläufigkeit.

Modularisierung: Ein klassischer Dialoggraph hat genau einen Start- und einen Enddialog. Weiterhin könnten einmal spezifizierte Unterdialoge an mehreren Stellen oder in mehreren Anwendungen wiederverwendet werden. Abbildung 6 zeigt als Beispiel einen Dialoggraphen zur Anmeldung an einem System mit einem Start- und 2 Endpunkten (erfolgreich, nicht erfolgreich).

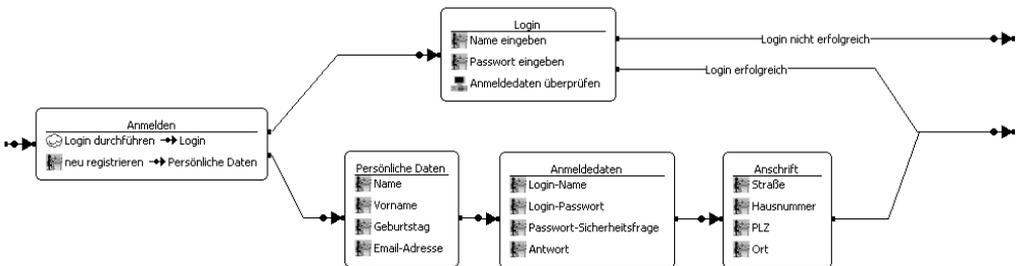


Abbildung 5: Dialoggraph Authentifizierung

Dieser Dialoggraph kann in mehrere Anwendungen bzw. an mehreren Stellen des Systems, an denen eine Authentifizierung notwendig ist, eingebettet werden. Alternativ kann dieser Teildialoggraph aber auch an den Anwendungskontext oder das Gerät angepasst oder voll-ständig ersetzt werden. Die Alternative (z.B. Authentifizierung über Fingerabdruckscanner) würde auf dem jeweiligen Endgerät dann anstatt der normalen Dialogabfolge dargestellt werden. Wichtig ist hierbei nur, dass alternative Dialoggraphen die gleiche Schnittstelle nach außen bereitstellen, in diesem Fall also 1 Eingangspunkt und 2 Endpunkte für die erfolgreiche bzw. nicht erfolgreiche Anmeldung.

4 Zusammenfassung und Ausblick

Wir haben in diesem Beitrag gezeigt, welche Möglichkeiten für die Anpassung von Dialogmodellen an verschiedene Geräte bestehen in der von uns benutzten Modellwelt existieren. Wir sind gerade im Rahmen eines größeren Projektes dabei, Dialogmodelle für verschiedene Rollen zu spezifizieren und deren Anpassung an unterschiedliche Geräte automatisch vornehmen zu lassen. Der Aufwand könnte damit bei n Rollen und m Geräten von $n*m$ auf n reduziert werden. Die Modularisierung von Teilen des Dialogmodells erlaubt Kapselung, Austausch und Wiederverwendung häufig benötigter Spezifikationen. Aufgrund der stetig wachsenden Gerätevielfalt wird in Zukunft eine Weiterentwicklung der gezeigten Techniken in Bezug auf Wiederverwendung erforderlich sein. Hier spielt die Identifikation und Anwendung von Patterns speziell für Aufgaben- und Dialogmodelle eine wichtige Rolle. Sie können einen die Spezifikationsarbeiten wesentlich erleichtern.

Literaturverzeichnis

- [BC04] Berti, S; Correani, F.; Mori, G.; Paternò, F.; Santoro, C.: A transformation-based environment for designing multi-device interactive applications. In: Proceedings of the 9th international conference on Intelligent user interfaces. Funchal, Januar 2004.
- [FR06] Forbrig, P.; Reichart, D. (2006): Modellbasierte Entwicklung modellbasierter Werkzeuge. Proc. 1. Workshop der Special Interest Group „Model-Driven Software Engineering“ der gasq. Hamburg, Dezember 2006.
- [LH00] Leung, K.R.P.H., Hui, L.C.K., Yiu, S.M., Tang, R.W.M.: Modeling Web Navigation by Statechart, Proc. COMPSAC'00.
- [LC03] Luyten, K., Clerckx, T., Coninx, K., Vanderdonckt, J.: Derivation of a dialog model from a task model by activity chain extraction. In Jorge, J., Nunes, N.J., e Cunha, J.F. (ed.), Proc. of DSV-IS 2003, LNCS 2844, Springer, 2003.
- [QT06] Quintero, R., Torres, V., Ruiz, M., Pelechanto, V.: A Conceptual Modeling Approach for the Design of Web Applications based on Services, Proc. ACM SE'06, Melbourne, USA, March, 2006.
- [RI06] Rieckhoff, S. (2006): Modularisierung von Navigationsspezifikationen unter Beachtung von HCI-Patterns, Diplomarbeit, Universität Rostock, 2006.
- [SE96] Schlungbaum, E.; Elwert, T.: Dialogue Graphs – A Formal and Visual Specification Technique for Dialogue Modelling. Springer, 1996.