

Werkzeug für aktivitätsbasierte Modellierung von Tätigkeiten

Gregor Buchholz, Peter Forbrig

Universität Rostock
18059 Rostock

{gregor.buchholz, peter.forbrig}@uni-rostock.de

Zusammenfassung

In Reengineering-Prozessen ist zur geeigneten Gestaltung des Zielsystems umfassendes Wissen über die zu unterstützenden Prozesse notwendig. Insbesondere für modellbasierte Neuentwicklungen kommt einer möglichst formalen Beschreibung der bisherigen Abläufe große Bedeutung zu; dies beinhaltet die Abstraktion von einzelnen Aktionen zu semantisch aussagekräftigen Aufgaben, die aus der Domäne analysiert wurden. Dieser Beitrag stellt eine werkzeuggestützte Methode vor, um aus Anwendungs-Protokollen von Altsystemen und während der Analyse generierten beispielhaften Aktions-Sequenzen Aufgabenmodelle zu erstellen, die als Grundlage für Design-Modelle des neu zu schaffenden Systems dienen.

1 Motivation

Informationen über Softwaresysteme, die über einen langen Zeitraum gewachsen sind und dabei schrittweise Erweiterungen in Funktionalität und unterstützten Tätigkeiten erfahren haben, sind oft schwer zu erheben. Sofern beispielsweise wegen eines angestrebten Architekturwechsels, der verwendete(n) Programmiersprache(n), der Anzahl der bei einer Neuentwicklung zu berücksichtigenden Altsysteme oder anderen Rahmenbedingungen eine Transformation allein auf Code-Ebene nicht durchführbar ist, ist eine Erhebung von Wissen höherer Abstraktion notwendig, um die so (wieder-)gewonnenen Anforderungen in das neue System zu überführen. Zahlreiche Methoden und Werkzeuge sind entstanden, um entsprechend den Entwicklungsparadigmen model-driven und model-based engineering (MDE, MBE) Aufgabenmodelle als wesentlichen Bestandteil der Softwareentwicklung vom Design über die Entwicklung von Oberflächen und Anwendungslogik bis hin zu Usability-Tests zu etablieren. Dem vorangestellt ist jedoch die Herausforderung, die aus der Analyse bestehender Systeme oder aus der Anforderungserhebung mittels Interviews mit Domain-Experten, Anwendern und Entwicklern gewonnenen Informationen in formale Modellstrukturen zu überführen. Ziel dieses Ansatzes ist daher die Erstellung eines Aufgabenbaumes mit temporalen Abhängigkeiten angelehnt an CTT [2] zwecks Reengineerings.

2 Abgrenzung

Das *ProM* Framework [4] bietet dank zahlreicher PlugIns vielfältige Unterstützung bei der Erstellung von Prozessmodellen aus aufgezeichneten Handlungs-Protokollen anderer Systeme. Auch das *RomanTutor*-System basierend auf dem *Sequential Pattern Mining* (SPM) [1] zielt auf die Erstellung von Modellen aus Protokollen. Im Gegensatz zu diesen und ähnlichen Systemen, die eine Hierarchisierung wenn überhaupt dann vorrangig aufgrund technischer Überlegungen vornehmen, liegt unser Fokus auf dem Erstellen hierarchischer Modelle, bei denen die Hierarchie eine Zerlegung der Aufgaben in Teilaufgaben aus Anwendersicht abbildet und dadurch das Verständnis aller Beteiligten fördern und abgleichen helfen soll.

3 Eigener Ansatz

3.1 Erstellung eines Analysemodells

Aus der Dokumentation der/s Altsysteme/s und Interviews mit Domain-Experten werden die Aufgaben des Systems auf abstrakter Ebene ermittelt. Ausgehend von der Wurzel stellen die darunterliegenden Ebenen Verfeinerungen dar, welche die darüberliegende Aufgabe zerlegen. Eine solche mittels entsprechender Werkzeuge erstellte Struktur dient als zentrales Artefakt der weiteren Schritte. Das als Client-Server-System implementierte Werkzeug erwartet ein Analysemodell als Hauptbestandteil eines Projektes. Darüber hinaus können die identifizierten Rollen hinterlegt werden, die an der Ausführung der Aufgaben beteiligt sind.

3.2 Erfassung von Aktivitäten

Eventuell vorhandene Protokolle des Altsystems können ebenso wie Interviews und Systemanalysen als Quelle zur Ermittlung einer Menge von Aktivitäten dienen, die zur Ausführung der Aufgaben notwendig sind. Die Zuordnung der Aktivitäten zu den zuvor modellierten Aufgaben kann im Werkzeug explizit stattfinden oder implizit durch das Generieren von Aktionssequenzen vorgenommen werden, welche die Erfüllung von Aufgaben durch Nacheinanderausführung von Aktionen beispielhaft darstellen. Dazu wird ei-

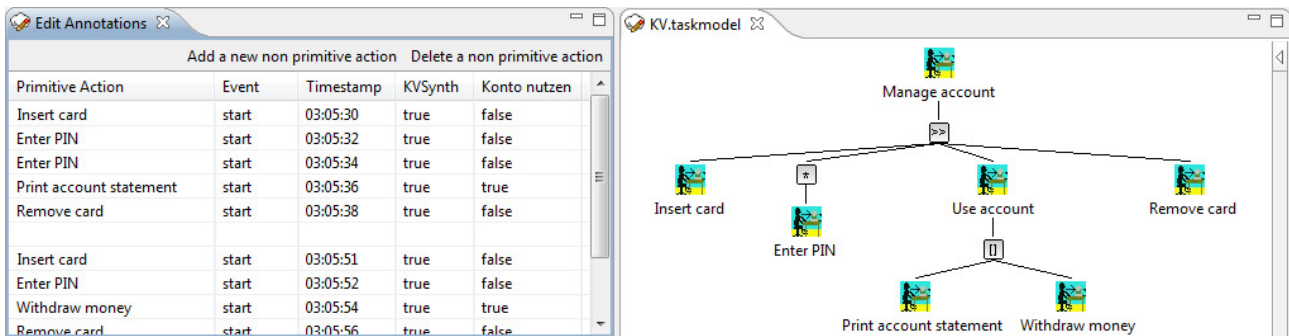


Abbildung 1: Aktivitäts-Protokolle und Aufgabenmodell

ne Teil-Aufgabe ausgewählt, mehrere Clients können sich unter Auswahl einer Rolle mit dieser Szenario-Instanz verbinden und gemeinsam die Aktionen ausführen. Jeder Client stellt dabei zunächst nur die der jeweiligen Rolle und Aufgabe zugeordneten Aktionen zur Auswahl - die übrigen können optional eingeblendet werden. Die so generierten Szenarien kommen zusammen mit importierten Aktivitäts-Protokollen aus anderen Quellen (auf Angabe der XSL-Datei wird hier aus Platzgründen verzichtet) im nächsten Schritt als Grundlage zur Erstellung der Baumstruktur zum Einsatz.

3.3 Synthese von Baumfragmenten, Mergen mit Analysemodell

In diesem Schritt werden zunächst die zu berücksichtigenden Szenarien ausgewählt. Es sind drei Algorithmen (PrefixSpan [3], Apriori und BIDE+) zur Mustererkennung implementiert. Ziel ist die Identifizierung zusammengehöriger Aktionssequenzen, die als Kandidaten für die Zusammenfassung zu einer Aufgabe vorgeschlagen werden. Eine Übersicht der gefundenen Sequenzen ermöglicht die Auswahl und Benennung der Sequenzen. Diese Bezeichnungen werden als neue Knoten im Baum zwischen den Aufgaben aus der Analyse und den zusammengefassten Aktionen eingefügt. Eine angepasste Version des *LearnModel*-Algorithmus' erstellt dann die vollständige Struktur des Baumes und leitet die temporalen Operatoren aus den Protokollen ab. Dabei kommt der Erkennung der Choice-Relation eine besondere Bedeutung zu, da durch sie neue Knoten entstehen können.

Abbildung 1 zeigt auf der linken Seite kurze Ausschnitte aus Protokollen von Szenarien. Aus der Analyse wurde im Beispiel die Aufgabe „Manage account“ manuell identifiziert und als zunächst einzige Aufgabe und Wurzel des Baumes definiert. Die Erkennung der Choice-Relation zwischen den Aktionen „Print account statement“ und „Withdraw money“ führte zur Einführung eines neuen Aufgabenknotens, der hier mit „Use account“ benannt wurde. Die Abbildung zeigt weiterhin die übrigen aus den Protokollen ermittelten temporalen Relationen.

In einem Projekt können zu einem Analysemodell

mehrere Vorschläge für Modelle zur Integration der aus den Protokollen synthetisierten Modellfragmente abgelegt werden. Alternative Versionen, die sich aus der Auswertung unterschiedlicher Szenarien oder aus unterschiedlichen Belegungen für die Parameter ergeben haben, können verglichen, simuliert und evaluiert werden. Ein iteratives Generieren von Szenarien, Erzeugen von Modellen und Evaluieren der Ergebnisse führt dann zu einem Aufgabenmodell, das für den Designentwurf des Neusystems zur Verfügung steht.

4 Ausblick

Die Werkzeugunterstützung soll insbesondere die umfangreiche Referenzierung von Analyseartefakten ermöglichen. So soll zu jedem Modellelement nachvollziehbar sein, aufgrund welcher Information und Überlegung es entstanden ist. Dies dokumentiert sowohl die manuell getroffenen Entscheidungen als auch die algorithmisch ermittelten Strukturen und Operatoren. Außerdem soll ein Import-Wizard als Hilfestellung beim Aufbereiten externer Protokolle zur Szenarienanalyse entstehen.

Literatur

- [1] P. Fournier-Viger, R. Nkambou, and E. M. Nguifo. A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems. In *Proceedings of the 7th Mexican International Conference on Artificial Intelligence: Advances in Artificial Intelligence, MICAI '08*, pages 765–778, Berlin, Heidelberg, 2008. Springer-Verlag.
- [2] F. Paterno. *Model-Based Design and Evaluation of Interactive Applications*. Springer-Verlag, London, UK, 1st edition, 1999.
- [3] J. Pei, J. Han, B. Mortazavi-Asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, and M.-C. Hsu. Mining sequential patterns by pattern-growth: The prefixspan approach. *IEEE Trans. on Knowl. and Data Eng.*, 16:1424–1440, November 2004.
- [4] B. van Dongen, A. de Medeiros, H. Verbeek, A. Weijters, and W. van der Aalst. The prom framework: A new era in process mining tool support. In G. Ciardo and P. Darondeau, editors, *Applications and Theory of Petri Nets 2005*, volume 3536 of *Lecture Notes in Computer Science*, pages 1105–1116. Springer Berlin / Heidelberg, 2005.