

Werkzeuggestützte interaktive Formalisierung textueller Anwendungsfallbeschreibungen für den Systemtest

Mario Friske, Holger Pirk
Fraunhofer FIRST
Kekuléstraße 7
D-12489 Berlin
{mario.friske,holger.pirk}@first.fhg.de

Abstract: Tester stehen oft vor der Aufgabe, Testfälle aus textuellen Anforderungsdokumenten zu erstellen. In der Regel sind diese natürlichsprachlichen Anforderungen nicht formal genug, um daraus automatisch Testfälle zu generieren. In diesem Papier stellen wir ein Werkzeug vor, mit welchem textuelle Anwendungsfallbeschreibungen interaktiv aufbereitet werden können, so dass die anschließende automatisierte Testfallgenerierung möglich wird. Des Weiteren werden die in dem Werkzeug verwendeten Metamodelle beschrieben.

1 Einleitung

Zu den wichtigsten qualitätssichernden Maßnahmen in der Softwareentwicklung zählt das Testen. Der Softwaretest sollte systematisch und spezifikationsbasiert durchgeführt werden, da nur so die Konsistenz zwischen Anforderungsspezifikation und Implementierung nachgewiesen werden kann. Insbesondere bei sich häufig ändernden Spezifikationen, welche typischerweise in iterativen Entwicklungsprozessen auftreten, ist aus Zeit- und Kostengründen eine Automatisierung des Testprozesses wünschenswert. Sofern diese auch das automatische Erstellen von Testfällen umfassen soll, sind Spezifikationen notwendig, welche formal genug sind, um als Ausgangspunkt für die Testfallgenerierung zu dienen.

Eine häufig genutzte Spezifikationsform sind Anwendungsfälle (Use Cases). Diese beschreiben typische Nutzer-System-Interaktionen, deren Ausführung einen bestimmten Nutzen für den Anwender des Systems hat. Anwendungsfälle lassen sich auf unterschiedliche Art und Weise notieren: Als Ganzes können Use Cases in UML Use Case Diagrammen visualisiert werden. Die detaillierte Beschreibung der Interaktionen kann sowohl in Form textueller Anwendungsfallbeschreibungen als auch durch Modelle erfolgen, wozu in der Regel UML-Interaktionsdiagramme verwendet werden [Coc00].

Vielfach werden die textuellen Anwendungsfallbeschreibungen Teil des Vertrages zwischen Auftraggeber und Auftragnehmer und damit nicht nur Grundlage für den Systementwurf, sondern auch für System- und Abnahmetests. Eine vollautomatische Überführung in Testfälle ist wünschenswert, aufgrund des Einsatzes natürlicher Sprache mit allen daraus resultieren Problemen, wie beispielsweise Mehrdeutigkeiten, jedoch nicht möglich.

Bei Fraunhofer FIRST wird mit dem *Use Case Validator (UCV)* ein Werkzeug entwickelt, welches die systematische Erstellung von Testfällen für den Systemtest aus textuellen Anwendungsfallbeschreibungen unterstützt. Mit dem auf dem *Eclipse Modeling Framework (EMF)* [EF] basierenden UCV können textuelle Anwendungsfallbeschreibungen interaktiv aufbereitet und in formalisierte Modelle überführt werden. Solche formalisierten Modelle lassen sich unter Verwendung der zugehörigen Metamodelle mit den Methoden und Werkzeugen der *Model Driven Architecture* [OMG03] automatisiert in Testfälle überführen. Dazu werden die formalisierten Anwendungsfallbeschreibungen über die Zwischenstufen *Plattformunabhängige Testfälle (PIT)* und *Plattformspezifische Testfälle (PST)* in ausführbare Testskripte transformiert, siehe Abbildung 1. Nachfolgend wird beschrieben, wie mit dem UCV der erste Transformationsschritt, die interaktive modellbasierte Aufbereitung von textuellen Anwendungsfallbeschreibungen, durchgeführt werden kann.

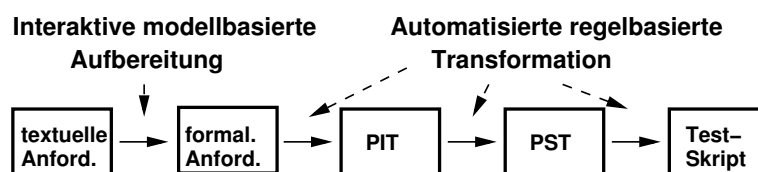


Abbildung 1: Testfallerzeugung aus textuellen Anforderungsbeschreibungen durch mehrstufige Modelltransformation

2 Interaktive Formalisierung für den Systemtest

Da textuelle Anforderungsdokumente typischerweise inkonsistent, mehrdeutig und unvollständig sind, lassen diese sich nicht in Modelle überführen, welche präzise genug sind, um als Ausgangspunkt für die modellbasierte Entwicklung zu dienen, ohne dass dabei schon Entwurfsentscheidungen getroffen werden. Die formalisierten Anforderungen sollten noch jede mögliche Systemrealisierung zulassen, welche die informell notierten Anforderungen des Auftraggebers an das System erfüllt. Die Formalisierung für den Systemtest muss diese Forderung nicht erfüllen, da nicht mehr alle möglichen Systemrealisierungen berücksichtigt werden müssen, sondern nur noch die konkrete Implementierung betrachtet wird. Informationen aus dem Systementwurf können bei der Formalisierung berücksichtigt werden [FS05].

Mit dem in Abbildung 2 dargestellten *Use Case Validator (UCV)* können Anwendungsfallbeschreibungen unter Berücksichtigung von Informationen aus dem Systementwurf interaktiv angereichert werden, um daraus anschließend Testfälle für den Systemtest zu generieren. Dabei wird die Art und Abfolge der Interaktionen betrachtet, was für viele Systeme, insbesondere geschäftsprozessunterstützende DV-Systeme, ausreichend ist.

Dazu werden zunächst die textuellen Anwendungsfallbeschreibungen in den UCV eingelesen. Während des Imports werden die einzelnen Abschnitte der Beschreibungen interaktiv typisiert. In der *Outline View* (1) werden alle importierten Anwendungsfallbeschreibungen überblicksartig dargestellt.

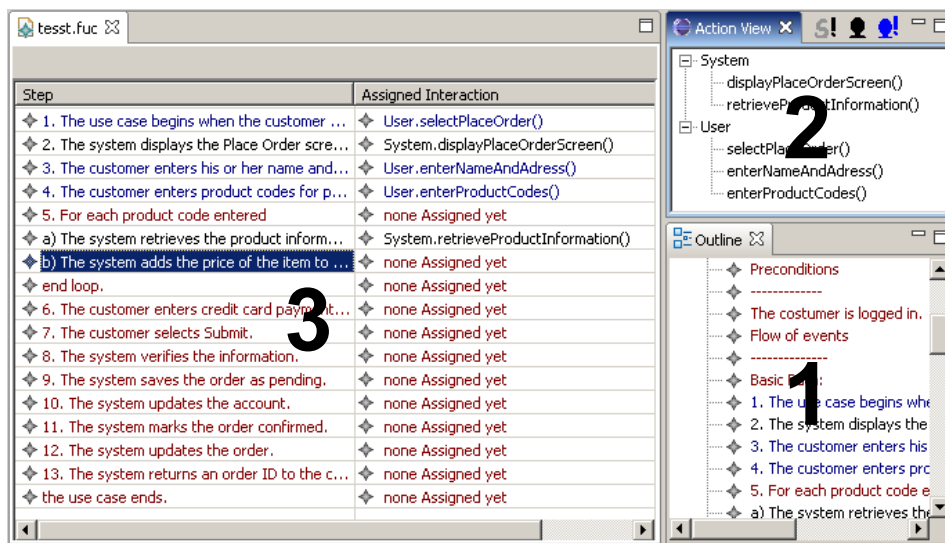


Abbildung 2: Ausschnitt der Benutzeroberfläche des UCV

Mögliche Nutzer-System-Interaktionen des zu testenden Systems werden in der *Action View* (2) baumartig dargestellt. Sofern diese Entwurfswahl schon vorliegt, kann sie importiert werden. Während des interaktiven Formalisierens kann diese ggf. erweitert und modifiziert werden. Ein komplettes Erstellen der Entwurfswahl direkt mit dem UCV ist ebenfalls möglich.

Der eigentliche Formalisierungsvorgang erfolgt mit dem *Use Case Editor* (3). Mit ihm werden die Schritte und Interaktionen markiert und verknüpft. Weiterhin wird der Kontrollfluss durch Auswählen von Kontrollflusselementen und Bestimmen der zugehörigen Blöcke formalisiert. Dabei wird schrittweise ein formales Modell der textuellen Anwendungsfallbeschreibung erstellt, in welchem die Mehrdeutigkeiten der textuellen Repräsentation interaktiv beseitigt wurden. Dieses Modell ist durch ein Metamodell definiert, welches im Folgenden beschrieben wird.

3 Die Metamodelle des Use Case Validator

Die Formalisierung der Anwendungsfälle geschieht unter Verwendung des UCV-Metamodells. Eine der Anforderungen an die Entwicklung des UCV war es, die textuellen Anwendungsfallbeschreibungen und die während der Formalisierung hinzugefügten Informationen getrennt zu halten. Daraus ergibt sich eine Teilung des Modells in drei Teile, die nachfolgend einzeln im Detail vorgestellt werden.

Das Metamodell der textuellen Anwendungsfallbeschreibung ist auf der linken Seite in Abbildung 3 dargestellt. Es beschreibt, dass ein Anforderungsmodell (*UCModel*) mehre-

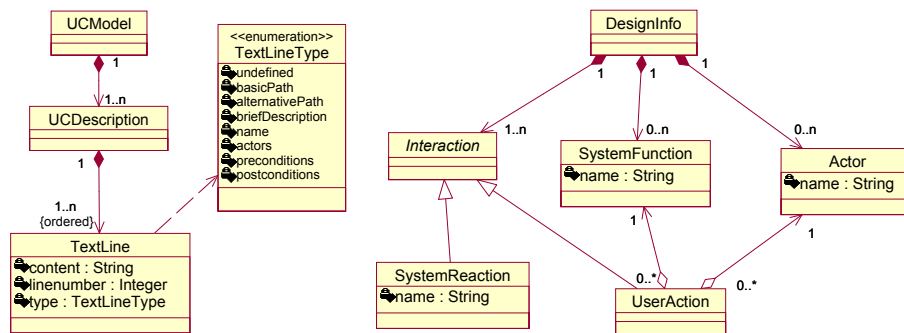


Abbildung 3: Metamodell der nichtformalisierten Anwendungsfallbeschreibungen und Metamodell der Entwurfsinformation

re textuelle Anwendungsfallbeschreibungen (*UCDescription*) beinhaltet. Diese bestehen jeweils aus einer Menge von geordneten Textzeilen (*TextLine*).

Anwendungsfallbeschreibungen sind in der Regel strukturierte Dokumente, welche aus einzelnen Abschnitten bestehen, z.B. Kurzbeschreibung, Vorbedingung, Hauptablauf, etc. Entsprechend ist die Bedeutung einer Textzeile abhängig vom Abschnitt, welchem sie angehört. Information über die Abschnittsart wird in einem Attribut als *TextLineType* gespeichert.

Auf der rechten Seite in Abbildung 3 ist das Metamodell der zur Formalisierung genutzten Informationen aus dem Systementwurf dargestellt. Die Entwurfsinformation (*DesignInfo*) beinhaltet Wissen über mögliche Nutzer-System-Interaktionen (*Interactions*), welche entweder von Nutzern aufgerufene Funktionen des Systems (*UserAction*) oder Reaktionen des Systems (*SystemReaction*) sind. *UserAction* fasst den Nutzer (*Actor*) und die durch ihn aufgerufene Funktion des Systems (*SystemFunction*) zusammen, was die Übersicht bei komplexen Systemen mit mehreren Akteuren und einer Vielzahl von Systemfunktionen erhöht. Es ist geplant, die Interaktionen um Parameter zu erweitern.

Das formalisierte Anforderungsmodell (*FUCModel*) stellt die Verbindung zwischen Anforderungsmodell (*UCModel*) und Entwurfsinformation (*DesignInfo*) her, siehe Abbildung 4. Dieses Metamodell beinhaltet sowohl Klassen, welche der Formalisierung der einzelnen Interaktionen dienen, als auch Klassen, um den Kontrollfluss zu modellieren. Das formalisierte Anforderungsmodell umfasst mehrere formalisierte Anwendungsfallbeschreibungen (*FUCDescription*). Diese bestehen aus Blöcken (*Block*), welche die eigentlichen Kontrollflusselemente darstellen.

Sequenzen (*Sequence*) ermöglichen es, mehrere Blöcke aneinanderzureihen. In den Schritten (*Step*) werden Interaktionen aus der Entwurfsinformation den Textzeilen der Anwendungsfallbeschreibungen zugeordnet. Komplexere Kontrollflusskonstrukte sind Schleifen (*Loop*) und Verzweigungen (*Switch*). Diese bestehen aus einer Bedingung (*Condition*) und einem Schleifenkörper, bzw. einem if- und einem else-Block. Die Bedingung kann bei der späteren Verarbeitung ausgewertet werden, um den Kontrollfluss im Testzenario zu ermitteln.

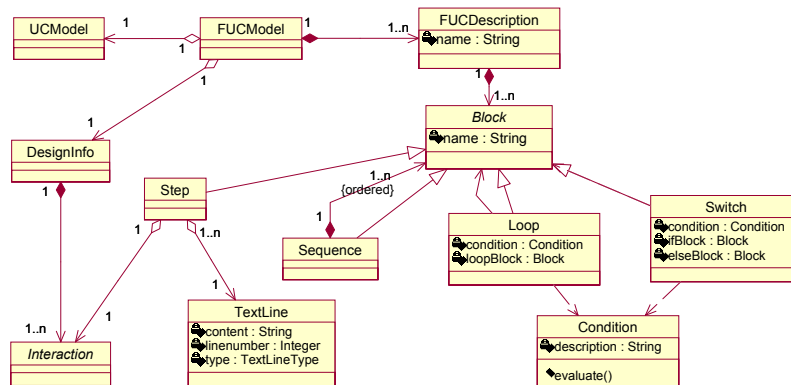


Abbildung 4: Metamodell der formalisierten Anwendungsfallbeschreibung

4 Zusammenfassung und Ausblick

Es wurde dargestellt, wie mit dem Use Case Validator (UCV) textuelle Anwendungsfallbeschreibungen in durch ein Metamodell definierte, formalisierte Anwendungsfallbeschreibungen interaktiv überführt werden können. Die erstellten Modelle sind formal genug, um Ausgangspunkt für die in [Fri04] beschriebene mehrstufige automatische Transformation von Anwendungsfallbeschreibungen in Testfälle zu sein.

Zur Zeit arbeiten wir an der Realisierung der nachfolgenden Transformationsschritte mithilfe EMF-basierter Transformationsmaschinen [DST]. Weiterhin ist vorgesehen, den UCV um die Behandlung von Interaktionsparametern zu erweitern.

Literatur

- [Coc00] Alistair Cockburn. *Writing Effective Use Cases*. Addison-Wesley, 2000.
- [DST] DSTC. Tefkat. <http://www.dstc.edu.au/Research/Projects/Pegamento/tefkat/>.
- [EF] The Eclipse Foundation. Eclipse Modeling Framework. <http://www.eclipse.org/emf/>.
- [Fri04] Mario Friske. Testfallerzeugung aus Use-Case-Beschreibungen. *Softwaretechnik-Trends*, Band 24, Heft 3, 2004.
- [FS05] Mario Friske und Holger Schlingloff. Von Use Cases zu Test Cases: Eine systematische Vorgehensweise. In Torsten Klein, Bernhard Rumpe und Bernhard Schätz, Hrsg., *Tagungsband des Dagstuhl Workshops "Modellbasierte Entwicklung eingebetteter Systeme" (MBEES)*. Technische Universität Braunschweig, 2005.
- [OMG03] Object Management Group. (MDA) Guide Version 1.0.1. <http://www.omg.org/mda/>, 2003.