

# Erkennung von Feature Interactions auf Nutzungsebene

## Modellierung und Verifikation von Abhängigkeiten

Johannes Grünbauer

Technische Universität München, Institut für Informatik

Lehrstuhl für Software und Systems Engineering, Boltzmannstr. 3, D-85748 Garching

14. Januar 2008

### 1 Motivation

Die Entwicklung von komplexen Softwaresystemen ist nach wie vor eine große Herausforderung. Die Funktionalität von eingebetteten und multifunktionalen Systemen, wie sie unter anderem im Automobil zu finden sind, nimmt stetig zu. Durch die Vielfalt der Funktionen wird die Entwicklung eines Gesamtsystems insofern erschwert, als durch die Verzahnung und Verbindung der einzelnen Funktionen ein komplexes und unüberschaubares Geflecht von Abhängigkeiten entsteht. Durch den Mangel eines globalen Verständnisses über das Verhalten des Systems aus Nutzersicht kann sich das Problem ergeben, dass Funktionszusammenhänge nicht immer erkannt werden und Funktionen sich gegenseitig in ihrer Ausführung stören können, wodurch es zu schwerwiegendem Fehlverhalten des Systems kommen kann. Werden derartige Fehler bei der Konzeption von Systemen erst sehr spät im Entwicklungsprozess entdeckt, ist es meist mit hohem finanziellem Aufwand verbunden oder gar unmöglich, diese Schwächen im Nachhinein zu bereinigen. Daher müssen im gesamten Software-Engineering Prozess, beginnend im Requirements-Engineering, die Zusammenhänge von Funktionen in ihrer ganzen Vielfalt und in den unterschiedlichen Ausprägungen verstanden werden.

Die gegenseitige Beeinflussung von Funktionen wird als *Feature Interaction* bezeichnet (vgl. [3]) und ist für funktionierende Systeme auch notwendig. Das Phänomen, dass durch die Nutzung von Funktionen unspezifizierte Seiteneffekte entstehen können, wird im Allgemeinen als *unwanted feature interaction problem* bezeichnet. Um dieses Problem zu vermeiden ist es notwendig, schon in der frühen Phase des Software Engineering Prozesses die Zusammenhänge zwischen Funktionen sichtbar zu machen.

### 2 Abhängigkeitsmodell

Nachfolgend wird ein Ansatz zur Erkennung von ungewollten Feature Interactions auf der Nutzungsebene präsentiert. Dafür wird eine neue Sicht auf ein System vorgestellt, die die Abhängigkeiten zwischen den Funktionen und den Zuständen eines Systems be-

schreibt. In einem Abhängigkeitsmodell werden die statischen Zusammenhänge innerhalb eines Systems abgebildet. Diese werden über drei grundsätzliche Modellierungselemente (Nutzungsfunktionen, Zustände und Beziehungen) angezeigt (siehe Abb. 1).

**Nutzungsfunktionen** sind im Abhängigkeitsmodell als Kreis dargestellt und repräsentieren die Funktionen, die der Benutzer eines Systems anwenden kann und durch deren Anwendung sich eine wahrnehmbare Veränderung des Systems zeigt. Die Anwendung einer Funktion kann zu einer Auswirkung auf den Modus einer Funktion (z. B. deren Anwendbarkeit), zum anderen auf Zustände des Systems haben.

Weiterhin werden in einem Abhängigkeitsmodell **Zustände** modelliert (dargestellt in Textform), die ein System bzw. dessen Umgebung annehmen kann. Diese werden innerhalb des Abhängigkeitsmodells auf Augenhöhe, also innerhalb einer einzigen Sicht dargestellt. Sie beschreiben die Gesamtheit der möglichen Zustände, die aufgrund der Nutzung des Systems eintreten können bzw. die Zustände der Umgebung, die die Funktionsweise eines Systems beeinflussen können. Zustände werden im Modell als Text dargestellt, zusammengehörige Zustände mit geschweiften Klammern gruppiert.

**Beziehungen** in einer Abhängigkeitssicht sind durch Pfeile repräsentiert und verbinden schließlich die Nutzungsfunktionen und Zustände miteinander und beschreiben den Einfluss dieser Modellierungselemente aufeinander. So wird durch Beziehungen beispielsweise dargestellt, ob eine Funktion durch eine andere zur Ausführung angestoßen werden kann oder ob durch die Anwendung einer Funktion bestimmte Systemzustände eingenommen werden können.

Durch diese Elemente werden die Zusammenhänge, die in den Anforderungsdokumenten beschrieben sind, modelliert und sichtbar gemacht.

In Abb. 1 ist ein Abhängigkeitsmodell dargestellt, welches den Zusammenhang zwischen der Nutzungsfunktion *Zündung* und der Nutzungsfunktion *Klimaanlage* beschreibt. Über die Beziehung *activate* zwischen *Zündung* und dem Zustand *Zündung ein* wird angezeigt, dass durch die Anwendung der Funktion

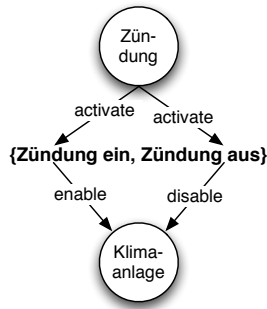


Abbildung 1: Abhängigkeitsmodell

*Zündung* der Zustand *Zündung ein* herbeigeführt werden kann. Analog dazu gilt die Herbeiführung des Zustands *Zündung aus* durch die Nutzungsfunktion *Zündung*.

Weiterhin wird durch die Beziehung *enable* beschrieben, dass durch den Zustand *Zündung ein* die Funktion *Klimaanlage* anwendbar gemacht werden kann (d. h. der Anwender bzw. das System kann die Funktion *nutzen*). Analog dazu wird durch den Zustand *Zündung aus* die Klimaanlage ausgeschaltet und die Nutzung dieser unterbunden.

Weitere Beziehungen, die in einem Abhängigkeitsmodell verwendet werden können sind *trigger*, *cancel* und *subfunction* (siehe [1]). Mit *trigger* wird angezeigt, dass eine Funktion automatisch zur Nutzung angestoßen wird, mit *cancel* wird ein Funktionsabbruch angezeigt. Die Beziehung *subfunction* zeigt eine hierarchische Gliederung einer Funktion in deren Teilfunktionalitäten an. Für die Modellierung der Systemfunktionalität auf Nutzungsebene sind die vorgestellten Beziehungen ausreichend und die Beschreibungen aus den Anforderungsdokumenten können vollständig in ein Abhängigkeitsmodell überführt werden.

### 3 Analyse des Abhängigkeitsmodells

Das Abhängigkeitsmodell zeigt sämtliche Beziehungen eines Systems zusammengefasst an. Darüber lässt sich jedoch noch keine direkt Aussage darüber ableiten, welche Beziehungen abhängig von welchen Zuständen gelten können. Beispielsweise ist es bereits aus einem intuitivem Verständnis heraus nicht möglich, dass die Nutzungsfunktion *Zündung* gleichzeitig die Zustände *Zündung ein* und *Zündung aus* herbeiführt. Daher ist es für die Analyse des Modells notwendig, so genannte *Instanzen* des Modells zu bilden. Eine Instanz eines Abhängigkeitsmodell beschreibt einen Zeitpunkt, zu dem bestimmte Zustände aktiv sind und ausgehend davon Beziehungen zwischen den Modellelementen gelten können. Dazu ist es zunächst notwendig, dass für die Beziehungen eine Semantik definiert ist und davon abhängig das Abhängigkeitsmodell formal beschreibbar ist. In [1] ist die Semantik der Beziehun-

gen beschrieben und wie die Abbildung in ein aussagenlogisches Modell erfolgt. Über spezielle SAT-Solver ist es schließlich möglich, nicht nur eine Lösung, sondern alle gültige Belegungen für ein Modell auszugeben. Jede gefundene Belegung beschreibt eine Instanz des Abhängigkeitsmodells.

Durch die Instanzenbildung können folgende Situationen analysiert und darauf aufbauend folgende Fragen beantwortet werden:

- Wenn zwei oder mehr Beziehungen auf eine Funktion einwirken, welche Auswirkungen sind für das System zu erwarten? (Beispiel: *enable/disable*-Kombination).
- Transitivität. Kann eine Funktion indirekt über dritte Funktionen beeinflusst werden?
- Was bedeutet es, wenn zwei Funktionen versuchen, unterschiedliche Zustände, die nicht gemeinsam auftreten dürfen, herbeizuführen.

Weiterhin können durch die Modellierung gezielt Fragen in der Art beantwortet werden,

- ob eine Funktion ausgeführt werden kann wenn ein bestimmter Zustand aktiviert ist oder
- welche Voraussetzungen grundsätzlich gelten müssen, dass eine Funktion ausgeführt werden kann.

## 4 Zusammenfassung

Durch den ständig wachsenden Funktionsumfang von eingebetteten Systemen ist es notwendig, bereits in einer frühen Phase des Software-Engineerings die Funktionszusammenhänge zu beschreiben und sichtbar zu machen. Die vorgestellte Methode macht es möglich, in den Anforderungsdokumenten unspezifiziertes Systemverhalten zu erkennen und Konflikte aufzulösen.

Das Thema der Erkennung von ungewollten Feature Interactions in der RE-Phase ist ein zentrales Thema innerhalb des Forschungsprojekts REMSES (siehe [2]). Ziel des vom BMBF geförderten Projekts ist es, einen validierten, praxistauglichen Leitfaden für systematisches Requirements-Engineering und -Management eingebetteter Systeme zu entwerfen. Anhand einer Fallstudie wird die hier vorgestellte Methode validiert.

## Literatur

- [1] Johannes Grünbauer. *Feature Interaction auf Nutzungsebene. Modellierung und Analyse der Abhängigkeiten*. PhD thesis, Technische Universität München, 2008 (to be published).
- [2] Webseite des REMsES-Projekts. <http://www.remses.org>.
- [3] Pamela Zave. FAQ Sheet on Feature Interaction. <http://www.research.att.com/~pamela/faq.html>, 2004.