

Ausführbare und testbare Spezifikationen mit xUML

Mit modellbasierten Ansätzen effizient und ökonomisch zu korrekter Software

Agile Methoden haben durchaus ihre Berechtigung, wer dabei jedoch die Mächtigkeit richtig genutzter Modelle ausser Acht lässt, verpasst ein zentrales Element eines modernen Softwareprozesses.

Wie werden Anforderungen stabilisiert

Präzise und lückenlose Pflichtenhefte sind in vielen Projekten noch immer eine Wunschvorstellung. Als Grund dafür werden oft mangelnde Zeit oder fehlende Unterstützung von Kundenseite ins Feld geführt. Ob man bei der Spezifikation wirklich an alles gedacht hat, wird sich eh erst mit ersten Prototypen zeigen.

Aus diesem Grund wird in Softwareprojekten oft sehr früh mit der Teilrealisierung des geplanten Systems begonnen. Statt die Anforderungen und das konzeptionelle Systemdesign zu präzisieren, wird bereits viel wertvolle Kapazität für technische Lösungen absorbiert. Das funktioniert solange das Entwicklungsteam die Erwartungen des Kunden trifft. Der Werkzeugkasten agiler Methoden mag zwar helfen das Risiko von Fehlentwicklungen zu reduzieren,

doch ab einer gewissen Projektgrösse stoßen auch diese Ansätze an ihre Grenzen.

Modellbasierte Ansätze

Wer effizient und ökonomisch Software entwickeln will, kommt nicht um modellbasierte Ansätze herum. Modelle sind heute mehr als eine Sammlung grafischer Repräsentationen im Sinne von „ein Bild sagt mehr als tausend Worte“. Die dargestellten Informationen und Zusammenhänge lassen sich maschinell interpretieren und so effizient im Entwicklungsprozess nutzen.



Der Autor Rolf Gubser ist Partner und Berater bei KnowGravity, Inc. KnowGravity ist ein führendes Beratungsunternehmen im Bereich der modellbasierten Entwicklung. rolf.gubser@knowgravity.com

Hier liegt übrigens auch ein Schlüssel im effektiven Umgang mit Modellen: man trennt die Implementation von der Spezifikation und pflegt nicht ein grosses Modell das schrittweise bis zum Abbild des Codes erweitert wird. Im Software-Engineering haben sich die folgenden Modelle etabliert:

- im plattformunabhängigen Modell werden Anforderungen in Form einer lösungsneutralen Spezifikation formuliert.
- das plattformspezifische Modell beschreibt die technische Umsetzung der Spezifikation auf der Zielplattform.

Beide Modelle werden mit der Notation der UML (Unified Modeling Language) erstellt. Darauf basierend lassen sich viele Aufgaben im Entwicklungsprozess automatisieren – bis hin zur vollständigen Code-Generierung. Anhand des einfachen MP3-Players „FLASHman“ möchte ich illustrieren, welche Schlüsselrolle lösungsneutrale Spezifikationen bei der modernen Entwicklung von Softwaresystemen spielen.

Lösungsneutrale Spezifikation

Eine lösungsneutrale Spezifikation ist ein wichtiges Bindeglied zwischen Kundenanforderungen und der technischen Implementation. Es handelt sich dabei nicht bloss um einen abstrahierten Entwurf, sondern um eine präzise und vollständige Beschreibung der Funktionalität des künftigen Systems.

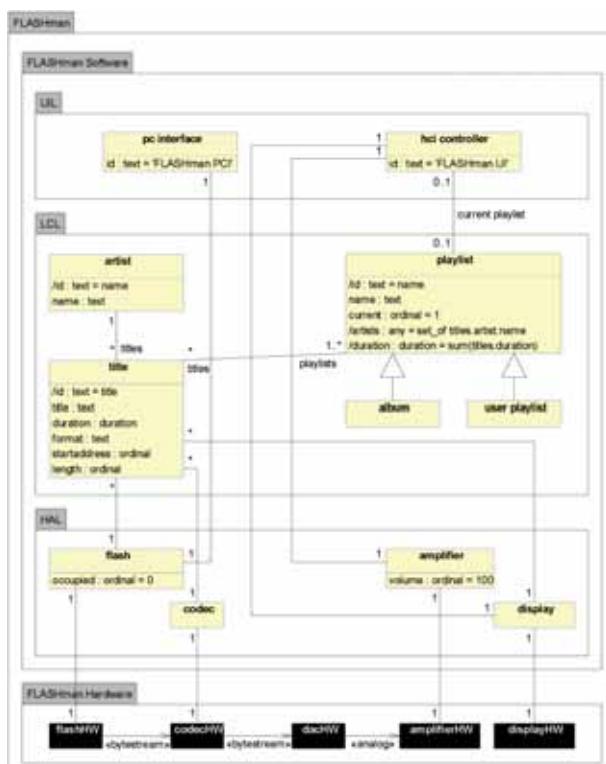


Abbildung 1 zeigt die Domänenobjekte am Beispiel FLASHman



all-electronics.de

ENTWICKLUNG. FERTIGUNG. AUTOMATISIERUNG



Entdecken Sie weitere interessante
Artikel und News zum Thema auf
all-electronics.de!

Hier klicken & informieren!





FAEL, Langackerweg 10, 5303 Würenlingen
Tel. (+41 44) 268 37 11, info@fael.ch, www.fael.ch

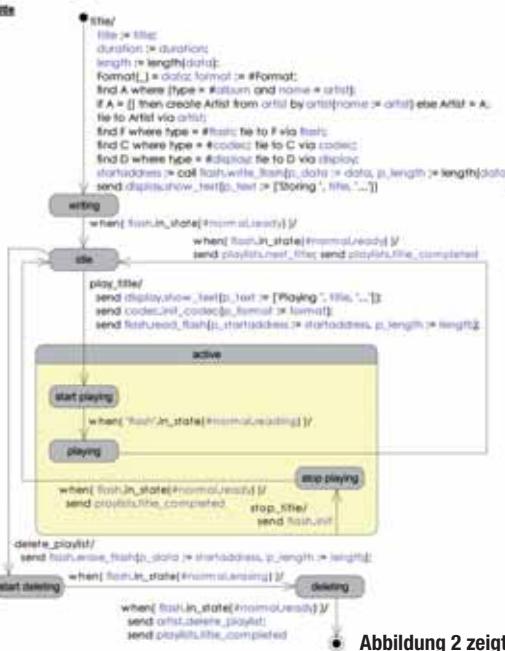


Abbildung 2 zeigt die Verwendung von OCL und Action Language im Zustandsdiagramm

Dabei zeigen Klassendiagramme die fachlichen Objekte und ihre strukturellen Eigenschaften. Mittels Use-Case- und einfachen Sequenzdiagrammen wird die Funktionalität definiert, die das System seinen Benutzern zur Verfügung stellt.

Schliesslich lässt sich mit Zustandsdiagrammen das dynamische Verhalten der fachlichen Objekte spezifizieren.

An dieser Stelle plädiere ich für die Verwendung der ausführbaren UML, xUML (executable UML) genannt. Die xUML ist eine präzise Variante der UML, die nebst der grafischen Notation auch Gebrauch von den textuellen Erweiterungen OCL (Object Constraint Language) und der UML Action Semantics macht. Damit wird es möglich, komplexe Aktionen und Berechnungen so zu beschreiben, dass die Funktionalität gleich ausführbar wird.

Involvierung der Kundenseite

Doch was sollen präzisere UML-Modelle, wenn die Kundenseite jetzt schon keine UML-Diagramme lesen will? Genau das ist der Punkt: Kunden lesen keine UML-Diagramme, sondern sie sollen direkt mit

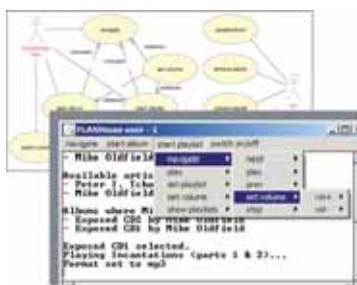


Abbildung 3 zeigt das simulierte User Interface des MP3-Players FLASHman.

Durch die Tatsache, dass die Spezifikation ausführbar ist, entsteht bei allen Beteiligten schnell ein gutes Gefühl bezüglich Vollständigkeit.

Modellbasiertes Testen

Auf dieser Grundlage lässt sich das Testen nun systematisieren: die auf der Simulation durchgespielten Testfälle und -daten lassen sich aufzeichnen und als Regressionstests beliebig wiederholen (insbesondere nach Änderungen an der Spezifikation). Eine gut ausgestattete

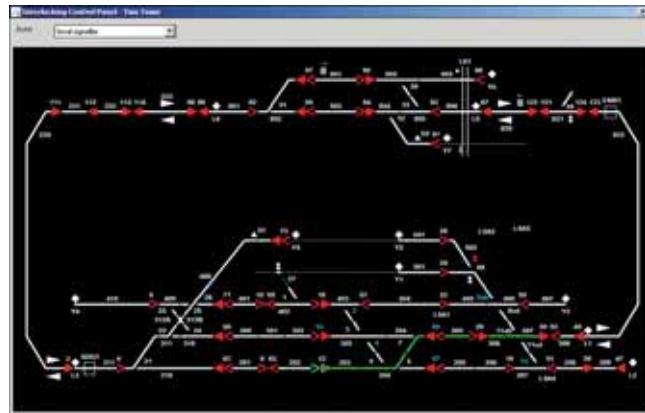


Abbildung 4 zeigt die ausführbare Spezifikation eines Stellwerksystems aus Benutzersicht

Simulationsumgebung erlaubt nebst Black-box- und Glas-Box-Tests auch eine feingranulare Kontrolle der Simulationszeit. Dies ist eine grosse Hilfe bei der Weiterentwicklung der Systemfunktionalität. Da die Tests direkt auf der Spezifikation erfasst und mit den Anforderungen verknüpft sind, vereinfacht sich das Test-Management massiv. Zum einen lässt sich die Spezifikation validieren und zum anderen können daraus auch Tests für die Implementation generiert werden.

Erfahrungen

Unabhängig davon, ob eine ausführbare Spezifikation zur externen Realisierung vergeben wird oder einem Code-Generator als Input dient, ist sie eine präzise Vorgabe, die von Mensch und Maschine eindeutig verstanden wird und sich entsprechend nutzen lässt.

Modellbasierte Ansätze skalieren sowohl für kleine als auch für grosse Systeme, wie zum Beispiel Stellwerksysteme im Bahnbereich, die auch durch mehrere Entwickler im Parallelbetrieb spezifiziert und getestet werden müssen. Ausführbare Spezifikationen sind der effizienteste Weg um Anforderungen rasch zu stabilisieren und präzise zu formulieren. Unsere Erfahrung zeigt, dass sich ein ausführbares xUML-Modell eines realen Systems innerhalb weniger Tage erstellen lässt. Darin sind jedoch die Aufwände für die Abklärung der geforderten Funktionalität nicht enthalten. Aber genau das ist der Punkt: diese Aufwände sind sowieso zu leisten, was in einem konventionellen Entwicklungsprozess später (oft durch die Programmierer) und damit wesentlich teurer geschieht. Dies ist am falschen Ort eingesetzte Agilität. (Rolf Gubser)

 infoDIREKT www.elektronikjournal.de

924eil1108

Veranstaltungskalender

Mobilität braucht Energie, aber welche?

Dienstag, 25.November 2008, 17:30 - 19:00,
Alost Konnex Gebäude, Brown Boveri Strasse 7, 5400 Baden,
Info und Anmeldung: www.fael.ch → Anlässe → Focus 531
Donnerstag, 27.November 2008, 17:30 - 19:00,
Gemeindesaal, Metzgplatz, 5600 Lenzburg,
Info und Anmeldung: www.fael.ch → Anlässe → Focus 532