Dissertationen

Christine Kohring: Ausführung von Anforderungsdefinitionen zum Rapid Prototyping

1. Referent: Prof. Dr.-Ing. Manfred Nagl
2. Referent: Prof. Dr. rer. nat. Gregor Engels
Datum der Prüfung: 12. Februar 1996

Zusammenfassung: Prototyping wird im Requirements Engineering dazu eingesetzt, die formulierten Systemanforderungen zu überprüfen. Insbesondere bei der Entwicklung von eingebetteten Systemen ist es wichtig, schon frühzeitig das dynamische und zeitliche Verhalten des angestrebten Systems anhand eines vorläufigen Modells zu validieren und Aspekte der Verteilung auf mehrere, möglicherweise unterschiedliche Zielrechner zu berücksichtigen.

Den Schwerpunkt der Arbeit bilden die Konzeption und Realisierung eines flexiblen und adaptablen Ausführungswerkzeugs, mit dessen Hilfe Anforderungsdefinitionen direkt interpretiert werden können. Auf diese Weise wird ein spezieller Prototyping-Ansatz verfolgt. Voraussetzung für die Entwicklung eines Interpreters für Anforderungsdefinitionen war zunächst die Existenz einer ausführbaren Modellierungssprache, die den besonderen Eigenschaften eingebetteter Systemen Rechnung trägt.

Bei der Entwicklung der integrierten Entwicklungsumgebung wurde besonderer Wert auf Flexibilität und Adaptabilität an geänderte Benutzeranforderungen gelegt. Flexibilität wurde dabei auf drei Ebenen definiert. So wurden bereits bei der Auswahl einer geeigneten Modellierungssprache (1) mehrere Varianten berücksichtigt. Diese wurden bei der Realisierung der integrierten Werkzeugumgebung (2) auf verschiedene Weise umgesetzt. Editor- und Analysewerkzeug können weitgehend aus der EBNF der Sprache generiert werden. Außerdem bietet die Werkzeugumgebung (3) die Möglichkeit einer verschränkten Vorgehensweise, so daß der Benutzer während der Ausführung erkannte fehlerhafte oder unvollständige Teile der Spezifikation sofort verbessern kann. Eine weitere Forderung an den RE-Interpreter lautete, (4) eine einfache Simulation der Zielsystemeigenschaften anzubieten.

Das Interpreterwerkzeug ist modular aufgebaut und umfaßt eine allgemein verwendbare Basisschicht, die verschiedenen Varianten der Modellierungssprache - eine integrierte SA/CM/IM-Sprache - unterstützt. Der zentrale Interpretationszyklus besteht aus dem globalen Netzflußinterpreter mit darin enthaltenen Interpretern für Kontroll- und Prozeßspekifikationen und einem Scheduler, der die Simulation einer verteilten Zielhardware erlaubt. Zur Überwachung und Auswerteung des Ausführungsvorgangs beinhaltet die RE-Entwicklungsumgebung Werkzeuge für Instrumentierung und Monitoring. Diese Anteile geben während der Ausführung Aufschluß über die aktuellen Laufzeitdaten, wie z.B. die Zustände der Prozesse, die Belegungen der Datenflüsse, die Warteschlangen der simulierten Prozessoren etc.

Andreas Zeller: Configuration Management with Version Sets – A Unified Software Versioning Model and its Applications

Anschrift: TU Braunschweig, Abteilung Softwaretechnologie, D-38092 Braunschweig

1. Referent: Prof. Dr. Gregor Snelting
2. Referent: Prof. Dr. Walter F. Tichy
Datum der Prüfung: 01.04.1997

Zusammenfassung: Das Software-Konfigurationsmanagement (SCM) kennt eine Vielzahl von Konzepten, die nur schwer miteinander zusammenarbeiten. Diese Arbeit stellt ein vereinheitlichtes Versionierungs-Modell vor, das Modell der *Versionsmengen.* Versionsmengen kennzeichnen Versionen, Komponenten und Konfigurationen durch *Feature-Terme* – Boolesche Terme über Ausdrücke der Art (*Eigenschaft*: *Wert*). *Feature-Logik,* ein etablierter Formalismus für Wissensrepräsentation und logische Programmierung, definiert Aufgaben und Konzepte des SCM. Die Ergebnisse sind wie folgt:

Vereinheitlichte Versionierung. Versionsmengen sind ein einheitlicher Formalismus, mit dem alle Dimensionen der Versionierung (Revisionen, Varianten, Arbeitsumgebungen) als auch Querbeziehungen integriert werden.

Integration von Änderungen und Revisionen. Konfigurationsbedingungen in Feature-Logik decken das gesamte Spektrum zeitlicher Versionierung ab – von der Strenge der versionsorientierten SCM-Modelle bis zur Kombinationsfreudigkeit der änderungsorientierten SCM-Modelle.

Konsistenzprüfung unter Mehrdeutigkeit. Feature-Logik bestimmt Eigenschaften und Konsistenz von Konfigurationen wie auch abgeleiteter Komponenten und beschreibt so, wie sich Eigenschaften im SCM-Prozeß fortpflanzen. Unstimmigkeiten werden auch bei unvollständigen und mehrdeutigen Konfigurationsbeschreibungen erkannt.

Das Modell der Versionsmengen wurde in einem experimentellen SCM-System namens ICE implementiert. Über ein virtuelles Dateisystem können Anwender beliebige Versionsmengen bearbeiten. Einzelne Versionen werden als Dateien angesprochen; Versionsmengen als ganzes können über Versions-Verzeichnisse oder über die wohlbekannte C-Präprozessor-Darstellung bearbeitet werden. Die bekannten SCM-Verfahren werden durch einfache Dateioperationen auf Versionsmengen effizient realisiert. Dies macht ICE zu einer universellen Plattform für SCM-Anwendungen und unterstreicht die praktische Bedeutung von Versionsmengen.

(Erhältlich über den Autor, zeller@acm.org)