



Fraunhofer Institut
Experimentelles
Software Engineering

RTLOpen Erfahrungsbericht Architekturprinzipien bei der Fa. Hofmann

Autor:
Thorsten Keuler (IESE)



IESE-Report Nr. 060.07/D
RTLOpen-Report 012/D
Version 1.0
1. September 2007

Eine Publikation des Fraunhofer IESE

**Eine Veröffentlichung des
RTLOpen-Projekts FKZ 01 IS C14**

Das diesem Bericht zugrunde liegende Vorhaben wurde mit Mitteln des Bundesministeriums für Bildung und Forschung unter dem Förderkennzeichen FKZ 01 IS C14 gefördert.

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft. Das Institut transferiert innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von
Prof. Dr. Dieter Rombach (geschäftsführend)
Prof. Dr. Peter Liggesmeyer
Fraunhofer-Platz 1
67663 Kaiserslautern

Abstract

In diesem Dokument werden die Erfahrungen dokumentiert die die Firma Hofmann mit den im Projekt „RTLOpen“ entwickelten Architekturkonzepten im Speziellen bei der Entwicklung einer Reifengleichförmigkeitsmessmaschine (RGM) gemacht haben.

Schlagworte: software development, architecture, open source, RTLOpen

Inhaltsverzeichnis

1	Einleitung	1
2	Architekturkonzepte und Erfahrungen	2
2.1	Systemarchitektur	2
2.2	Erfahrungen	2
2.3	Softwarearchitektur	3
2.4	Architekturkonzepte für Echtzeitsysteme	3
2.5	Stile für Echtzeitsysteme	4
2.6	Erfahrungen	4
2.7	Sichten für Echtzeitsysteme	5
2.8	Erfahrungen	6
	Referenzen	8

1 Einleitung

Das Projekt RTLOpen (Open Realtime Linux für den Maschinenbau) entwickelt eine Referenzplattform, die flexibel, hoch performant und zukunftssicher ist. Sie basiert auf Open-Source-Komponenten wie beispielsweise dem Betriebssystem Linux. Die RTLOpen Plattform ist in einem Handbuch dokumentiert, das aus verschiedenen Teilen besteht. Ein Hauptbestandteil des Handbuchs ist eine angepasste Methode die zur systematischen Entwicklung von Echtzeitsystemen im Maschinen- und Anlagenbau dient. In diesem Dokument werden die Erfahrungen der Firma Hofmann mit Architekturprinzipien dokumentiert.

Das Projekt ist Teil der Deutschen Forschungsinitiative „Software Engineering 2006“ (SE-2006) des Bundesministeriums für Bildung und Forschung. Die Projektnummer ist 01 IS C 14.

2 Architekturkonzepte und Erfahrungen

Hier wird zunächst ein Überblick über die eingeführten Architekturkonzepte gegeben. Dabei gliedern sich Erfahrungen, Eindrücke und Einschätzungen als jeweilige Bewertung an die vorgestellten Bestandteile der Architekturkonzepte.

2.1 Systemarchitektur

Die Systemarchitektur beschreibt die Komponenten die das Gesamtsystem ausmachen und deren Zusammenhänge. Auf der Ebene der Systemarchitektur sind alle Systembestandteile unabhängig von ihrer Realisierung definiert. So findet man Komponenten die später durch Hardware umgesetzt werden als auch solche die von Software umgesetzt werden. In einer nachgeschalteten Entwicklungsphase entscheidet sich welche Systembestandteile wie umgesetzt werden.

2.2 Erfahrungen

Auf Systemarchitekturebene ergibt sich für Hofmann die in Abbildung 2-1 dargestellte Struktur. Sie wurde aus der Implementierung der Maschinensoftware automatisiert rekonstruiert. Sie zeigt im Wesentlichen die anvisierte Systemarchitektur, die aus den Messdatenerfassungssystemen in Echtzeit (COM_Module) und den Datenvisualisierungen über TCP (TCP_Module) besteht.

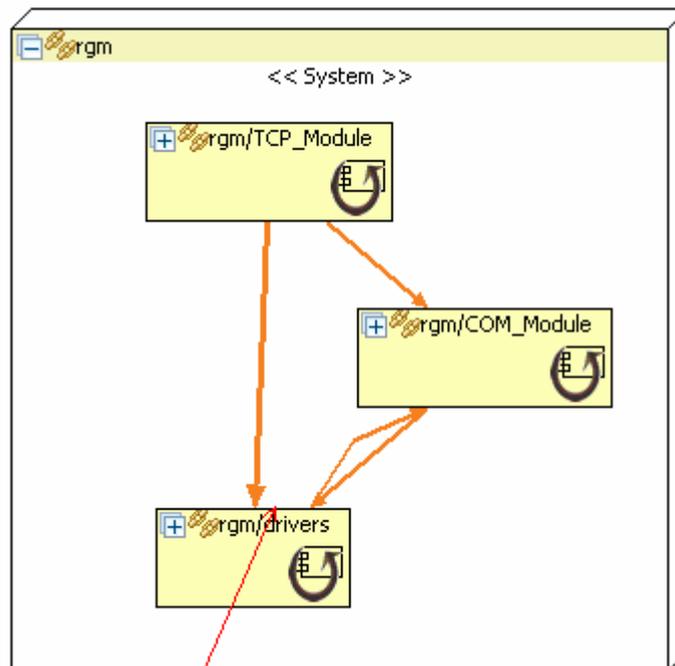


Abbildung 2-1 Systemdarstellung der Implementierung einer RGM von Hofmann

2.3 Softwarearchitektur

Die Softwarearchitektur bildet das Rückgrat für den Bau erfolgreicher softwareintensiver Systeme. Die Softwarearchitektur ist zum großen Teil verantwortlich dafür, dass Qualitätsattribute eines Systems wie z.B. Echtzeit, Performanz oder Zuverlässigkeit ermöglicht oder verhindert werden. Die Architektur stellt damit eine Investition dar, ein abstraktes, wieder verwendbares Modell, das von einem System auf das nächste übertragen werden kann. Die Architektur wird darüber hinaus als gemeinsames Kommunikationsmittel zwischen den Interessen der Systembeteiligten genutzt und ist der Bereich, in dem widersprüchliche Ziele und Anforderungen geklärt werden. Die richtige Architektur ist somit der Dreh- und Angelpunkt für den Erfolg eines Softwareprojekts.

2.4 Architekturkonzepte für Echtzeitsysteme

Eine Softwarearchitektur zu beschreiben, bedeutet, alle Informationen zusammenzuführen, die für alle Interessen der Systembeteiligten relevant sind, damit man dieselben Entwurfs- und Entwicklungsregeln befolgen kann. Nach der IEEE Standard Definition besteht die Beschreibung einer Architektur aus einer Sammlung von Architektur-Sichten. Um eine optimale Menge von Sichten in Bezug auf Einheitlichkeit und Konsistenz zu erhalten, müssen mehrere Schritte unternommen werden. Der erste Schritt ist die Identifizierung aller beteiligten

Stakeholder und die Zuordnung der für jeden Stakeholder wichtigen Anliegen. Danach kann jeder Belang analysiert werden und in Bezug auf eine einzelne Sicht dargestellt werden. Die Konsolidierung der Belange, die von verschiedenen Systembeteiligten angesprochen werden, und damit die Sichten, wird Architektur-Sicht genannt. Eine Architektur-Sicht wird abgeleitet von einer von drei grundlegenden Sichtarten: Modul-Sicht-Typ, Komponenten- und Konnektoren-Sicht-Typ oder Zuordnungs-Sicht-Typ. Es gibt mehrere Architekturstile, die einer dieser Sichtarten entsprechen. Die wichtigsten und deren Anwendung aus dem Bereich der Echtzeitsysteme werden im folgenden Abschnitt behandelt.

2.5 Stile für Echtzeitsysteme

Stile und Muster in der Softwarearchitektur sind wichtige Entscheidungsträger bei vorhergehenden Projektphasen. Sie können sowohl bestimmte Rahmenbedingungen implizieren als auch exakte Definitionen für Artefakte, Prozesse und Subaktivitäten und Kommunikationsbasen für alle Systembeteiligten. Während Architekturstile abstrakte Prinzipien definieren, an denen die Softwarearchitektur ausgerichtet ist, zeigen Architekturmuster (engl. pattern) konkretere Lösungen für immer wiederkehrende Probleme.

In der Domäne der Echtzeitsysteme können bestimmte Stile und Muster verwendet werden, die ihre Effektivität in vergangenen Projekten unter Beweis gestellt haben. Die Unterschiedlichkeit der Anforderungen an jedes System in diesen Domänen führt jedoch immer zu projektspezifischen Entscheidungen, die in Bezug auf die Architektur getroffen werden müssen.

Stile sind das abstraktere Konstrukt in Bezug auf die Softwarestruktur. Sie erfassen grundlegende Relationen, die in allen Phasen der Entwicklung realisiert werden müssen. Der Architekt ist verantwortlich dafür, dass alle Aussagen, Einschränkungen und Aspekte, die von einem Architekturstil erfasst werden, auch propagiert und überprüft werden.

2.6 Erfahrungen

Bei der Firma Hofmann hat sich im Bereich der Systementwicklung der RGM (Reifengleichförmigkeitsmessmaschine) ein Event-based Stil etabliert. Das bedeutet, das komplette Systemverhalten wird zur Laufzeit über Nachrichten getriggert die die Subprozesse der Maschine ansteuern. Der Stil ist allerdings nicht explizit dokumentiert und es gibt keinerlei Dokumentation wichtiger Entwurfsentscheidungen.

2.7 Sichten für Echtzeitsysteme

Die Softwarearchitektur lässt sich nach Kruchten [Kru+95] prinzipiell in 4+1 Sichten unterteilen, die wichtige Aspekte aus verschiedenen Blickwinkeln seitens der Stakeholder darstellen.

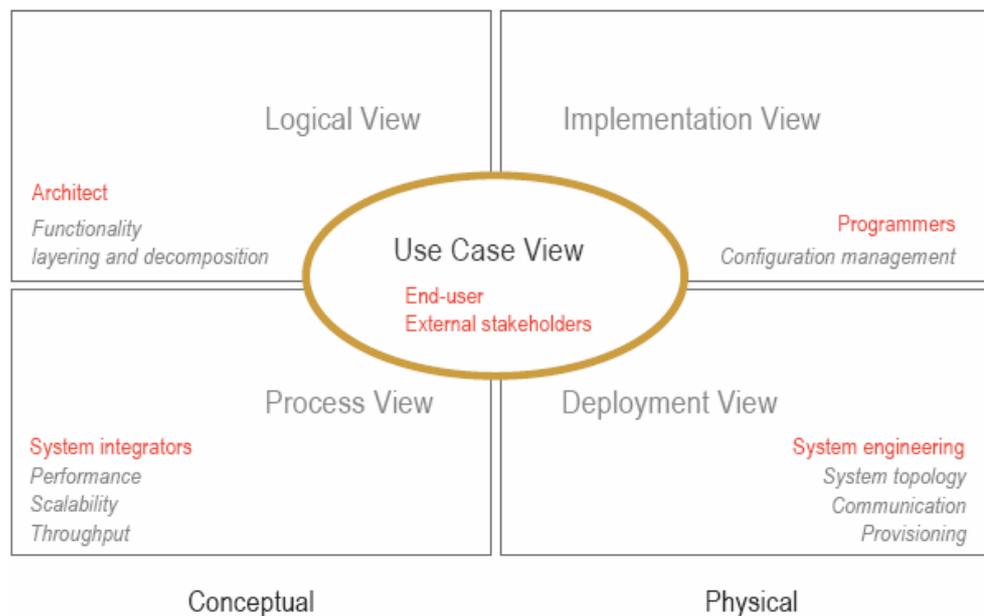


Abbildung 2-1 Grundlegende Sichten der Softwarearchitektur nach Kruchten [Kru+95]

Die logische oder konzeptuelle Sicht beschreibt ein Softwaresystem aus der Sicht seiner funktionalen Eigenschaften. Diese Sicht ist eng mit dem Bereich des Problems und der Domäne verknüpft und bildet somit eine wichtige Sicht für die Kommunikation mit Experten der jeweiligen Domäne. Auf der konzeptuellen Ebene ist es äußerst wichtig, jegliche Implementierungsentscheidung außen vor zu lassen.

Die Modulsicht zeigt alle verwendeten Module und ihre Relationen. Aufbauend auf der zugrunde liegenden Architektur (z.B. geschichtet oder Subsystem), sind sie so organisiert, dass sie den Einschränkungen des Architekturstils folgen. Die Modulsicht wird oft auch als strukturelle Sicht bezeichnet.

Die Prozess- oder Ausführungssicht berücksichtigt auch nicht-funktionale Aspekte (z.B. Performanz). Die Ausführungssicht kann auf jede Art von Systemabstraktion angewendet werden und kann damit direkt hierarchisch von der konzeptuellen Sicht abgebildet werden. Die Prozesssicht wird häufig auch als Verhaltenssicht bezeichnet.

Die physikalische Sicht bildet die Software auf die Hardware ab und fokussiert hauptsächlich auf Themen wie Performanz, Durchsatz, Verfügbarkeit und Skalierbarkeit. Diese Sicht beschreibt vor allem das Abbilden von Modulen auf ihre Laufzeitsprechungen und verfeinert somit den in der Ausführungssicht definierten Kontrollfluss.

2.8 Erfahrungen

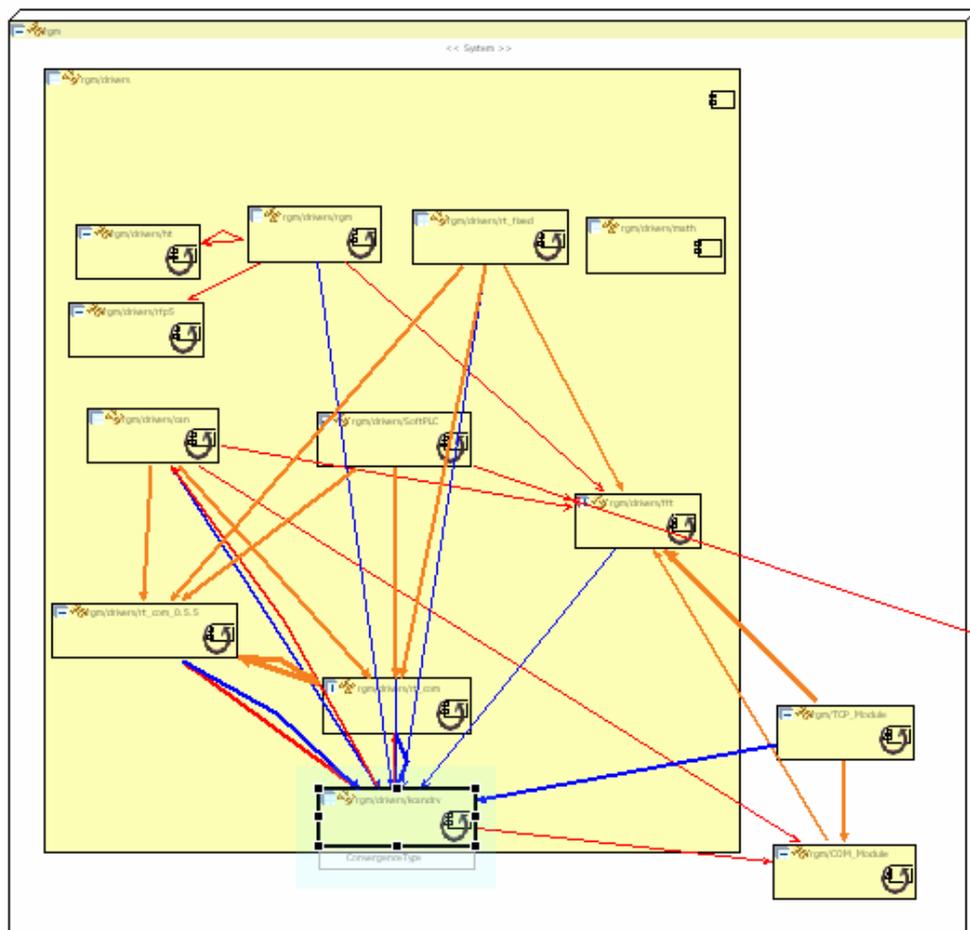


Abbildung 2-2 Modulsicht auf das implementierte RGM System

Sichten sind bei der RGM von Hofmann nicht explizit dokumentiert, jedoch konnte aus der Struktur der Implementierung der modulare Aufbau extrahiert werden (s. Abbildung 2-2). Sichtbar wird dabei, dass es einige Module gibt, die Abhängigkeiten zu fast allen anderen Modulen aufweisen. Gründe dafür sind unter anderem in der Art der Wiederverwendung von bereits erstellten Systemteilen zu finden. Das heißt, wird eine neue Maschine entwickelt, so erfolgt keine systematische Analyse der vorhandenen Artefakte, sondern bestehende Sys-

teme werden kopiert und speziell auf den neuen Kontext angepasst. Sinnvoll wäre folglich die Erstellung eines klaren Architekturmodells im Kontext der RGM-Anforderungen und darauf aufbauenden Sichten.

Abschließend lässt sich festhalten, dass Architekturkonzepte bei der Firma Hofmann lediglich implizit behandelt werden. Eine klare Fokussierung auf essentielle Sichten und deren Pflege sind aus Kosten/Nutzen-Sicht sehr empfehlenswert.

Referenzen

- [1] Elbertzhager, Kalmar, Keuler, „RTLOpen Methodendokument“, 2006

Dokumenten Information

Titel: RTLOpen Erfahrungsbericht
Architekturprinzipien bei der Fa.
Hofmann

Datum: 1. September 2007
Report: IESE-060.07/D
Status: Final
Klassifikation: Öffentlich

Copyright 2007, Fraunhofer IESE.
Alle Rechte vorbehalten. Diese Veröffentlichung darf für kommerzielle Zwecke ohne vorherige schriftliche Erlaubnis des Herausgebers in keiner Weise, auch nicht auszugsweise, insbesondere elektronisch oder mechanisch, als Fotokopie oder als Aufnahme oder sonstwie vervielfältigt, gespeichert oder übertragen werden. Eine schriftliche Genehmigung ist nicht erforderlich für die Vervielfältigung oder Verteilung der Veröffentlichung von bzw. an Personen zu privaten Zwecken.