

Eine Fallstudie zur Spezifikation von Fachkomponenten eines Informationssystems für Virtuelle Finanzdienstleister – Beschreibung und Schlussfolgerungen

Peter Fettke, Peter Loos, Markus von der Tann

Oktober 2001

Technische Universität Chemnitz
Professur Wirtschaftsinformatik II
{peter.fettke|loos|markus.vdtann}@isym.tu-chemnitz.de

2. Workshop "Modellierung und Spezifikation von Fachkomponenten"
GI-Arbeitskreis 5.10.3 "Komponentenorientierte betriebliche Anwendungssysteme"
5. Oktober 2001, Universität Bamberg
Eine Veranstaltung im Rahmen der Fachtagung „Verteilte Informationssysteme auf der
Grundlage von Objekten, Komponenten und Agenten“ (vertIS)

- **Einführung in die Fallstudie**
- **Überblick über die Fachkomponenten von cofis.net**
- **Ergebnisse der Fallstudie**
- **Zusammenfassung**

Ziele der Fallstudie

- **Spezifikation der Fachkomponenten des Systems cofis.net auf Basis des Memorandums**
- **Darstellung der Erfahrungen und Erkenntnisse**
- **Formulierung von Empfehlungen, in welche Richtung das Memorandum weiterentwickelt werden sollte**

Charakterisierung des Systems cofis.net

■ cofis.net: Prototyp eines Virtuellen Finanzdienstleisters

■ Definition Virtueller Finanzdienstleister

Ein Finanzdienstleister, der mit Hilfe moderner Informations- und Kommunikationstechnologien die Dienstleistungen am Markt agierender Anbieter von Finanzdienstleistungen vermittelt, neu kombiniert und konfiguriert.

■ Realisierte Produkte

- Virtuelle Überweisung
- Virtuelle Anlage

■ Implementierungstechnologien

- Java 1.1.8
- Brokat Twister 2.3.5
- Oracle 8i

Fachkomponenten von cofis.net (Auswahl)

- Benutzer- und Rechteverwaltung
- Kontenverwaltung
- Virtuelle Anlageverwaltung
- Bankleitzahlen
- 7 weitere Fachkomponenten

Bewertung der Syntax-Ebene

■ Vorteile

- Spezifikation lässt sich auf Basis der OMG IDL leicht vornehmen
- Durch Verwendung standardisierter Datentypen wird die Komponente unabhängig von ihrer Implementierung spezifiziert

■ Nachteil

- Manuelle Konvertierung nicht OMG IDL-konformer Datentypen

■ Beispiel

- Middleware Twister stellt untypisierte Datentypen bereit, die nicht IDL-konform spezifizierbar sind

Empfehlungen für die Syntax-Ebene

- Klärung, wie Spezifikation zu erfolgen hat, wenn verwendete IDL nicht konform zur OMG IDL ist
- evtl. Berücksichtigung alternativer Spezifikationsmethoden

Bewertung der Verhaltens-Ebene

■ Vorteil

- Notation zwingt den Ersteller, Invarianten sowie Vor- und Nachbedingungen eines Dienstes zu explizieren

■ Nachteile

- Formulierung des Verhaltens mit OCL ist prinzipiell schwierig durchzuführen, obgleich das Verhalten z.T. trivial ist.
 - fehlendes UML-Modell
 - teilweise umständliche Formulierungen notwendig

■ Beispiel

```
Bankleitzahlen::sucheBankleitzahl(name : bankname) : blz  
pre : name ist ein gültiger Bankname
```


Empfehlungen für die Verhaltens-Ebene

- Aussagen treffen, auf welche Art und Weise ein UML-Modell einzuführen ist, um die OCL ordnungsgemäß verwenden zu können
- Einführung einer sekundären allgemeinsprachlichen Notation, welche die angegebenen Verhaltensregeln zusätzlich erläutert

Bewertung der Abstimmungs-Ebene

■ Vorteile

- Nutzungsmöglichkeiten der Dienste einer bzw. mehrerer Fachkomponenten werden offenbart
- Abhängigkeiten zwischen Diensten verschiedener Komponenten werden deutlich

■ Nachteile

- Ähnlich der Verhaltens-Ebene: hohe Komplexität und schwere Verständlichkeit im Verhältnis zum Gehalt der Aussage

■ Beispiel

- Abstimmungs-Ebene der Komponente „*Virtuelle Anlage*“

VirtualInvestment

```
confirmInvestment (selectedVirtInvestment) or  
rejectInvestment (selectedVirtInvestment) or  
prologInvestment (selectedVirtInvestment) or  
cancelInvestment (selectedVirtInvestment) or  
payInterest (selectedVirtInvestment) or  
setExpirationDate (selectedVirtInvestment, expDate)
```

implies

```
Framework : createXMLDocument (InvestmentData)
```

before

```
FDL-Bus : transmitInvestmentData (InvestmentXMLData)
```

■ Spezifizierte Aussage

Die Dienste zum Beenden, Verlängern, Bestätigen und Ablehnen einer virtuellen Anlage bzw. der Zinszahlung benötigen jeweils die Dienste des Frameworks zum Erzeugen der XML-Dokumente und den FDL-Bus zum anschließenden Übertragen dieser.

Empfehlungen für die Abstimmungs-Ebene

- **Anführung einer Literaturquelle, die gut in temporale Logiken einführt**
- **Erstellung von umfassenden Beispielen zur Nutzung der temporalen Erweiterung von OCL**
- **Systematische Ergänzung der Notation um natürlichsprachliche Konzepte**

Bewertung der Qualitäts-Ebene

■ Nachteile

- Unklarheiten hinsichtlich
 - zu spezifizierender Größen
 - Vergleich dieser Größen
 - Methoden zur Erhebung

- erstellte Spezifikation hat einen „konstruierten“ Charakter

■ Beispiel: Spezifikation der Fachkomponente „Bankleitzahlen“

Spezifikationsbeispiel Qualitäts-Ebene

Systemumgebung:

- Prozessorarchitektur: Intel Pentium III 866 MHz
- Hauptspeicher: 1 GB RAM
- Windows NT 4.0
- Oracle 8i
- Brokat Twister 2.3.5

Unter dieser Systemumgebung können die Dienste der Komponente folgendermaßen spezifiziert werden. Es wird der Dienst “istGueltigeBankleitzahl” spezifiziert. Als Eingabedaten werden zufällig ermittelte, gleichverteilte Werte verwendet.

Qualitätseigenschaft	Spezifikation
Durchsatzzeit	Der Durchsatz beträgt 1,8 s bei einer Arbeitslast von 1000 Dienstanforderungen.
Antwortzeit	Die Antwortzeit beträgt 18 ms.
Antwortzeitverhalten	Das Antwortzeitverhalten beträgt 0,0324 ms.
Verfügbarkeit	keine Angabe
Wiederanlaufzeit	keine Angabe

Empfehlungen für die Qualitäts-Ebene

- **Konkretere Ausarbeitung, in welchem Umfang Qualitätsmaße zu spezifizieren sind und welche Verfahren zur Anwendung kommen sollen**
- **Einführung einer konkreten Notation unter Berücksichtigung eventueller Vorarbeiten in der Literatur**
- **Überlegungen hinsichtlich der Verwendung von Referenzimplementierungen**

Bewertung der Domänen-Ebene

■ Vorteile

- Einfacher Aufbau eines Begriffssystems auf Basis des vorgeschlagenen Thesaurus-Konzeptes
- Zentralisierung der Begriffsdefinitionen erwies sich als nützlich

■ Nachteile

- Die vorgeschlagene Normsprache konnte aufgrund fehlender Erläuterungen der Technik zur genauen Verwendung nicht eingesetzt werden
- Bisheriger Ansatz sieht keine komponentenübergreifende Terminologiespezifikation vor, dies erwies sich jedoch als nützlich

Empfehlungen für die Domänen-Ebene

- **Aufzeigen der Nutzungsmöglichkeiten und der Nützlichkeit von Fachnormsprachen an einem konkreten Anwendungsbeispiel**
- **Überlegungen hinsichtlich einer komponentenübergreifenden Terminologiepflege unter Berücksichtigung komponentenspezifischer Definitionen**

Allgemeine Aspekte

- Unklarheiten hinsichtlich der Spezifikation der Dienste von Komponenten-Anwendungs- und Komponenten-System-Frameworks
- Angaben zur Installation von Komponenten erscheinen hilfreich
- Spezifikation möglicher Konfigurationen von Komponenten (Customizing)
- Layout-Richtlinien zur Formatierung der Spezifikation

- Spezifikationen der Syntax- sowie Domänen-Ebene lassen sich unproblematisch erstellen
- Bekannte Zielkonflikte zwischen Präzision und Verständlichkeit der Aussagen auf Verhaltens- und Abstimmungs-Ebene
- Qualitäts-Ebene ist ausbaufähig