

Martin Engelen/Kai Bender (Hrsg.)

# GeNeMe98

Gemeinschaften in Neuen Medien

TU Dresden, 1./2.10.1998



JOSEF EUL VERLAG

Lohmar · Köln

Reihe: Telekommunikation und  
Mediendienste

Band 2

Herausgegeben von Prof. Dr. Dr. h. c. Norbert Szyperski, Köln, Prof.  
Dr. Udo Winand, Kassel, Prof. Dr. Dietrich Seibt, Köln, und Prof. Dr.  
Rainer Kuhlen, Konstanz

Doz. Dr.-Ing. habil. Martin Engelen  
Dipl.-Inf. (FH) Kai Bender (Hrsg.)

# GeNeMe98

Gemeinschaften in Neuen Medien

TU Dresden, 1./2.10.1998



**JOSEF EUL VERLAG**  
Lohmar · Köln

Die Deutsche Bibliothek – CIP-Einheitsaufnahme

**GeNeMe <1998, Dresden>:**

GeNeMe 98 : Gemeinschaften in neuen Medien / Technische Universität Dresden, Fakultät Informatik, Institut für Informationssysteme, Dozentur „Entwurfmethoden und Werkzeuge für Anwendungssysteme“. Martin Engelen; Kai Bender (Hrsg.). – Lohmar ; Köln : Eul, 1998.

(Reihe: Telekommunikation und Mediendienste ; Bd. 2)  
ISBN 3-89012-632-4

© 1998

Josef Eul Verlag GmbH

Brandsberg 6

53797 Lohmar

Tel.: 0 22 05 / 91 08 91

Fax: 0 22 05 / 91 08 92

e-mail: eul.verlag.gmbh@t-online.de

Alle Rechte vorbehalten

Printed in Germany

Druck: Rosch-Buch, Scheßlitz

**Gedruckt auf säurefreiem und 100% chlorfrei gebleichtem  
Papier**



Technische Universität Dresden  
Fakultät Informatik • Institut für Informationssysteme  
Dozentur „Entwurfsmethoden und Werkzeuge für Anwendungssysteme“

Doz. Dr.-Ing. habil. Martin Engelen  
Dipl.-Inf. (FH) Kai Bender  
(Hrsg.)

*Dresden, 1./2. 10. 1998*

# **GENEME98**

*Gemeinschaften in Neuen Medien*



*Workshop zu Organisation, Kooperation und Kommunikation  
auf der Basis innovativer Technologien*

*Forum für den Dialog zwischen Wissenschaft und Praxis zur  
Inversion der Virtualität (Ubiquitous Computing)*

unter der Schirmherrschaft von:

**Dr. W. Vehse**  
Staatssekretär für Wirtschaft  
des Landes Sachsen

**Prof. Dr. A. Mehlhorn**  
Rektor der TU Dresden

sowie unter Mitwirkung der  
GI-Regionalgruppe Dresden

und mit freundlicher Unterstützung folgender Partner:



IST priv. Institut für angewandte Software-  
Technologie GmbH, Dresden  
eine Ausgründung der TU Dresden auf dem  
Gebiet der Technologien und Anwendungen  
in den Neuen Medien



Heyde AG,  
Bad Nauheim/ Dresden  
Beratung • Software • Integration



## **B.2. Entwurfsmuster für verteilte Anwendungssystem-Architekturen**

*Dipl.-Wirtsch.-Inf. C. Hammel*

*Dipl.-Inf. M. Schlitt*

*Dipl.-Kfm. S. Wolf*

*Universität Bamberg*

### **1 Motivation**

In den letzten Jahren werden verstärkt neue Organisationsformen für Unternehmen diskutiert (vgl. [Zimm97], [DaMa93], [Bull+97]). Diese reichen von innerbetrieblicher Dezentralisierung der Entscheidungsbefugnis bis hin zur Bildung autonomer Geschäftseinheiten, die in virtuellen Unternehmen kooperieren. Neben der stärkeren Verteilung der Entscheidungsbefugnis stehen neue Formen der Arbeitsverteilung sowie die Tendenz zur Globalisierung im Vordergrund. Daraus ergeben sich weitreichende Anforderungen an betriebliche Informationssysteme. Diese betreffen insbesondere technische Merkmale wie Flexibilität und Verteilbarkeit von Anwendungssystemkomponenten sowie die Verfügbarkeit und Performanz des Gesamtsystems. Aus betriebswirtschaftlicher Sicht müssen die Zielverfolgung, die Konsistenz sowie Aspekte der Datensicherheit berücksichtigt werden (vgl. [FeSi+97]). Konventionelle Anwendungssysteme, die auf Datenintegration basieren, erfüllen insbesondere die Anforderungen nach Flexibilität und Verteilbarkeit nicht in ausreichendem Maße. Daher ist es notwendig, Anwendungssystem-Architekturen zu entwickeln, welche die Realisierung unterschiedlicher technischer und organisatorischer Verteilungsformen unterstützen. Dies erfordert einen integrierten Ansatz zur Modellierung verteilter Anwendungssysteme im Kontext organisatorischer und technischer Rahmenbedingungen. Eine geeignete Modellierungsmethode sollte somit Verteilungskonzepte für alle angesprochenen Aspekte eines betrieblichen Systems beinhalten.

Objektorientierte Methoden zur Unternehmensmodellierung bieten wesentliche Vorteile gegenüber konventionellen Ansätzen, da hiermit Struktur und Verhalten von Systemen über alle Betrachtungsebenen spezifiziert werden können [FeSi98]. Neuere Vorschläge sehen den Ausgangspunkt der Systemgestaltung auf der betriebswirtschaftlichen Ebene, die beispielsweise in Form von Geschäftsprozeßmodellen dargestellt wird [Wolf97].

## 2 Ein Ansatz zur Modellierung verteilter betrieblicher Systeme

### 2.1 Ebenenarchitektur eines verteilten Systems

Bislang scheint keine einheitliche Begriffsdefinition für ein „Verteiltes System“ in der Literatur vorzuliegen ([Lori88, 3], [Diet94, 3f]). In Anlehnung an [Ensl78] kann ein verteiltes System allgemein durch folgende Merkmale beschrieben werden (vgl. hierzu ausführlich [Diet94]):

- Aus strukturorientierter Sicht besteht ein verteiltes System aus mehreren (mindestens 2) Komponenten, die über Interaktionskanäle miteinander gekoppelt sind.
- Aus verhaltensorientierter Sicht sind die Komponenten autonom und kooperieren im Rahmen einer gemeinsamen Aufgabe.

Die Methodik des Semantischen Objektmodells (SOM) nutzt einen Architekturrahmen, um ein betriebliches System als verteiltes, offenes und sozio-technisches System auf mehreren Modellebenen zu spezifizieren (vgl. ausführlich [FeSi98, 176ff]). Im folgenden stehen die Architekturebenen der Geschäftsprozeßmodelle und der Ressourcenspezifikation (vgl. Abbildung 1) im Vordergrund.

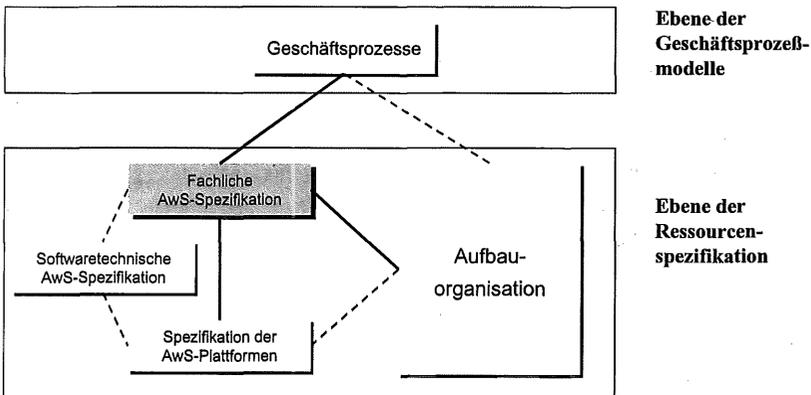


Abbildung 1: Architekturebenen der SOM-Methodik (in Anlehnung an [FeSi+98])

- Die Aufgabenebene eines betrieblichen Systems wird als System verteilter, kooperierender Geschäftsprozesse spezifiziert. Die Interaktionen zwischen Geschäftsprozessen werden durch die Koordinationsprinzipien Verhandlung (nicht-hierarchisch) und Regelung (hierarchisch) beschrieben. Struktur- und verhaltensorientierte Sichten auf das verteilte Geschäftsprozeßsystem dienen der Komplexitätsbewältigung.

- Verteilte Anwendungssysteme (AWS) werden als Ressourcen für die Durchführung automatisierter Aufgaben der Geschäftsprozesse spezifiziert. Diese Spezifikation wird weiter in Fachkonzept- und Softwarekonzeptspezifikation sowie die Spezifikation der AWS-Plattformen unterteilt. Nicht-automatisierte Aufgaben werden von personellen Aufgabenträgern (Aufbauorganisation) durchgeführt. Anwendungsneutrale Plattformen spezifizieren Rechner- und Kommunikationssysteme für verteilte Anwendungssysteme. Plattformen stellen abstrakte Maschinen z.B. in Form von Applikations-Middleware wie CORBA bereit.

Die Trennung von Aufgaben- und Aufgabenträgerebene ermöglicht eine auf fachliche Aspekte ausgerichtete AWS-Architektur für verteilte Systeme. Auf der Ebene der Ressourcenspezifikation wird der Einfluß von Verteilungsmerkmalen sowohl durch die Plattformspezifikation (technische Merkmale) als auch durch das Modell der Aufbauorganisation (organisatorische Merkmale) berücksichtigt.

## 2.2 Spezifikation der Aufgabenebene eines verteilten Systems

Die SOM-Methodik spezifiziert die Aufgabenebene eines verteilten Systems in Form von Geschäftsprozessen unter Nutzung wesentlicher Merkmale des objektorientierten Paradigmas [FeSi98, 183]. Modellbestandteile sind Objekte, die Zustände und Operatoren kapseln und Transaktionen, die Interaktionskanäle zwischen Objekten darstellen. Objekte und Transaktionen sind hierarchisch zerlegbar, was eine schrittweise Verfeinerung des Systementwurfs ermöglicht.

In Abbildung 2 ist der Geschäftsprozeß „Kapitalbeschaffung“ in Struktur- und Verhaltenssicht dargestellt. Aufgabe des Geschäftsprozesses ist es, Finanzierungsleistungen, die ein Unternehmen an seine Kunden erbringt, zu refinanzieren. Die *Kapitalbeschaffung* wird dazu von internen Kunden (hier: *Abwicklungssystem*) beauftragt, eine bestimmte Refinanzierungsart am Kapitalmarkt zu beschaffen. Zwischen den beteiligten Objekten werden nun Protokolle unter Nutzung von Transaktionen spezifiziert, welche die Koordination im Rahmen der Beschaffung und Übergabe der *Refinanzierungsleistung* an den internen Kunden beschreiben.

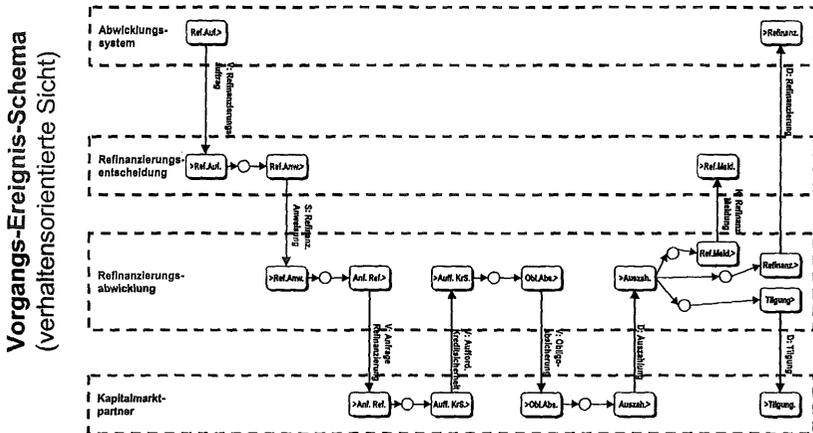
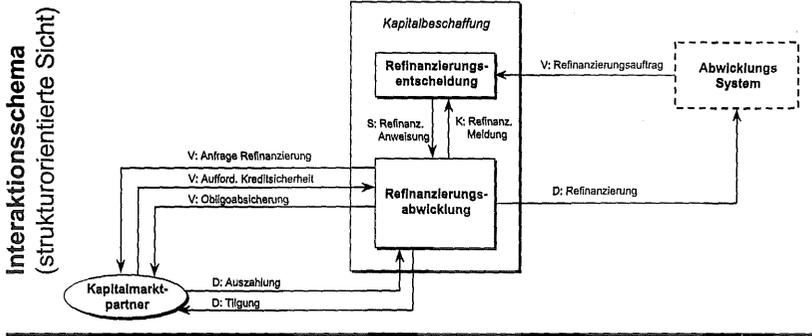
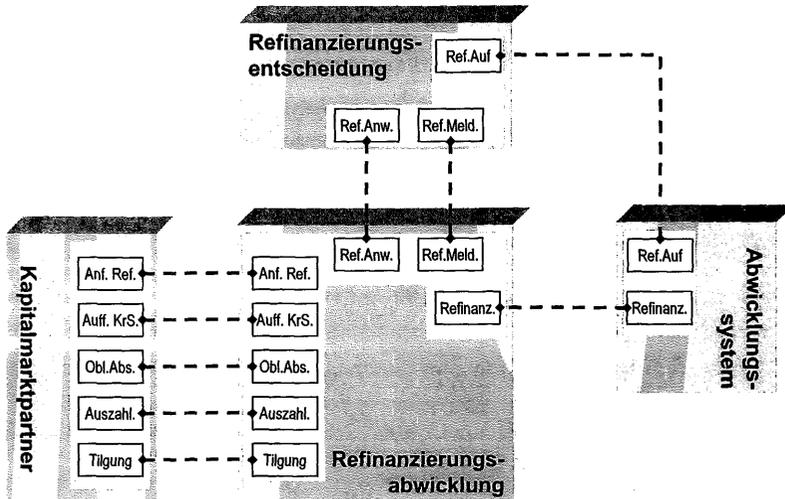


Abbildung 2: Sichten des GP-Modells „Kapitalbeschaffung“

### 2.3 Fachliche Spezifikation verteilter Anwendungssysteme

Die fachliche Spezifikation der Anwendungssysteme erfolgt in der SOM-Methodik auf der Grundlage eines Geschäftsprozessmodells. Für jedes Anwendungssystem wird ein konzeptuelles Objektschema (KOS), welches die strukturorientierte Sicht auf das AwS darstellt, sowie ein korrespondierendes Vorgangs-Objekt-Schema (VOS) als verhaltensorientierte Sicht entwickelt [FeSi98]. Um ein verteiltes System zu realisieren, erscheinen derart feingranulare Objektschemata als Spezifikationsform nicht optimal geeignet [Raue96, 40f].



**Abbildung 3: AWS-Komponenten des Geschäftsprozesses „Kapitalbeschaffung“**

Im folgenden wird daher ein Ansatz skizziert, die Verteilung eines Anwendungssystems anhand von Application Objects zu spezifizieren (vgl. [FeSi+97]). Dazu wird aus jedem betrieblichen Objekt ein Application Object abgeleitet. Die Beziehungen zwischen den so entstehenden Anwendungssystemkomponenten sind durch die Transaktionen zwischen den betrieblichen Objekten des Geschäftsprozeßmodells determiniert. Jedes Application Object spezifiziert den zu automatisierenden Teil der Aufgaben eines Geschäftsprozesses. In Abbildung 3 sind die AWS-Komponenten der Kapitalbeschaffung in ihrer Außensicht dargestellt.

Die Außensicht der Application Objects beschreibt die semantische Schnittstelle unabhängig von konkreten Plattformen. Erst nach der Auswahl einer konkreten Plattform für das verteilte System ist eine Realisierungsform für die Application Objects zu bestimmen. Die Verwaltung der Schnittstellen wird durch die möglichen Integrationsformen auf der gewählten Plattform determiniert.

Die Spezifikation eines verteilten Systems unter Nutzung von Application Objects erfolgt zunächst unabhängig von der Aufbauorganisation, die in Abstimmung mit dem Geschäftsprozeßmodell beschrieben wird. Anschließend werden organisatorische Einflüsse auf den Entwurf des verteilten Systems berücksichtigt.

### **3 Einfluß organisatorischer und technischer Rahmenbedingungen auf verteilte AWS-Architekturen**

Bei der Entwicklung verteilter Anwendungssysteme sind unterschiedliche Rahmenbedingungen zu berücksichtigen. Diese sollen im folgenden identifiziert und näher untersucht werden.

#### **3.1 Geschäftsprozeßinduzierte Rahmenbedingungen**

Ausgangspunkt für die Spezifikation eines Anwendungssystems ist eine hinreichend detaillierte Spezifikation der Aufgabenebene eines verteilten Systems. In der SOM-Methodik werden alle Aufgaben in einem korrespondierenden Geschäftsprozeßmodell erfaßt (vgl. Abschnitt 2.2 ). Auf Basis eines solchen Geschäftsprozeßmodells können Objektschemata für Anwendungssysteme abgeleitet werden [FeSi98, 199ff]. Die resultierenden Objektschemata beschreiben ein Anwendungssystem aus fachlicher Sicht mit maximaler Verteilung der Aufgaben [FeSi+97, 41]. Fachliche Einflußfaktoren determinieren die maximal mögliche Verteilung und spannen damit den Lösungsraum auf.

#### **3.2 Organisatorische Rahmenbedingungen**

Neben Anwendungssystemen werden in der SOM-Methodik personelle Aufgabenträger als Ressourcen für die Durchführung von Aufgaben betrachtet. Nicht alle im Geschäftsprozeßmodell spezifizierten Aufgaben eines verteilten betrieblichen Systems sind automatisierbar oder sollen automatisiert werden [Krum97, 128ff]. Nicht zu automatisierende Aufgaben werden vollständig personellen Aufgabenträgern zugeordnet, teilautomatisierte Aufgaben werden von personellen Aufgabenträgern und Anwendungssystemen in Kooperation durchgeführt [FeSi98, 178]. Diese gemeinsame Durchführung von Aufgaben macht es erforderlich, daß bei der Spezifikation von Anwendungssystemen die Aufbauorganisation berücksichtigt wird.

Um Aufbauorganisationen beschreiben zu können, werden im folgenden vier Merkmale herangezogen. Diese basieren im wesentlichen auf den Aktionsparametern zur Gestaltung der Aufbauorganisation nach GROCHLA [Gro82, 96ff].

Das Merkmal der **Arbeitsteilung** gibt an, nach welchen Kriterien die Aufgaben einer Unternehmung gegliedert werden. Eine Gliederung nach dem Verrichtungsprinzip faßt gleichartige Verrichtungen an unterschiedlichen Objekten zusammen und führt zur funktionalen Organisation. Eine Gliederung nach dem Objektprinzip bündelt unterschiedliche Verrichtungen an gleichartigen Objekten und führt zur divisionalen

Organisation (Spartenorganisation). In der Regel sind Mischformen, wie z.B. die Matrixorganisation, zu beobachten.

Das Merkmal der **Koordination** beschreibt die Verteilung der Entscheidungsbefugnisse innerhalb der Unternehmung. Als Ergebnis der Verteilung sind zwei Extrema denkbar: 1) die extreme Entscheidungscentralisation, bei der die Unternehmensführung jede Entscheidung übernimmt und 2) die extreme Entscheidungsdezentralisation, bei der alle Entscheidungen an hierarchisch untergeordnete Stellen übertragen werden. In Bezug auf dieses Merkmal sind in Unternehmen fast ausschließlich Mischformen zu finden.

Bestandteile des Merkmals der **Konfiguration** sind sowohl die Gliederungstiefe, d.h. die Anzahl der Hierarchieebenen der Unternehmung, als auch die Gliederungsbreite oder Leitungsspanne.

In neueren Organisationsformen wie z.B. virtuellen Unternehmen ist eine zunehmende Tendenz zur Globalisierung erkennbar [Bull+97, S.42]. Aus diesem Grund wird zusätzlich das Merkmal der **räumlichen Verteilung** von Organisationseinheiten herangezogen.

Zwischen den dargestellten Merkmalen existieren vielfältige Wechselwirkungen. Zum Beispiel ermöglichen leistungsfähige Mechanismen der Koordination breitere Leitungsspannen und damit den Abbau von Hierarchieebenen [Gro82, 108]. Bei der Gestaltung von Anwendungssystemen werden die Ausprägungen der organisatorischen Merkmale als gegeben angenommen; aus diesem Grunde werden auch deren Interdependenzen nicht weiter betrachtet.

### 3.3 Technische Rahmenbedingungen

Bei der Spezifikation von verteilten Anwendungssystemen müssen die Verteilungsaspekte der existierenden oder zu wählenden Plattformen berücksichtigt werden. Um die hohe Komplexität zu bewältigen, die die große Zahl unterschiedlicher verfügbarer Plattformen induziert, wird auf das Konzept der Integrationsformen von Anwendungssystemen ([Fers92, 14ff], [FeSi98, 216ff]) zurückgegriffen, um eine Klassifikation von Plattformen zu ermöglichen. Diese Integrationsformen stellen Lösungskonzepte für die Aufgabe der Automatisierung von Informationssystemen bereit und nehmen insbesondere auf das Formalziel der Integration Bezug. Die Integration umfaßt Einzelziele, die für die Entwicklung verteilter Anwendungssysteme von hoher Bedeutung sind (vgl. Abschnitt 0). Dabei handelt es sich um Plattformunabhängigkeit, die strukturorientierten Ziele Kontrolle der Redundanz, Beherrschung der Kommunikationsstruktur, sowie die verhaltensorientierten Ziele Gewährleistung von Konsistenz und Verfolgung der Ziele eines Anwendungssystems

([Fers92, 13f], [FeSi98, 214ff]). Aus Sicht des vorliegenden Beitrags relevante Integrationsformen werden im folgenden beschrieben<sup>31</sup>.

### **Datenintegration**

Ein datenintegriertes AwS [Fers92, 17] faßt Datenbestände und Datenkanäle in Form eines konzeptuellen Schemas zusammen. Auf dem dadurch definierten Zustandsraum operieren Funktionen. Da alle Funktionen auf demselben Speicher arbeiten, spricht man von enger Kopplung der Funktionen. Wichtige Teilsysteme der Anwendungssystem-Architektur sind Daten- und Funktionsmodell. Zahlreichen Anwendungssystemen, die in der Literatur beschrieben oder der Praxis eingesetzt werden, liegt das Konzept der Datenintegration zugrunde (vgl. z.B. [Mert93], [Scheer91]).

### **Objektintegration**

Ein objektintegriertes AwS umfaßt Objekte, die über ein Kommunikationssystem Nachrichten austauschen [Fers92, 21ff]. Im Gegensatz zur Datenintegration liegt hier eine lose Kopplung der Komponenten vor. Die Spezifikation erfolgt beispielsweise in der SOM-Methodik anhand eines konzeptuellen Objektschemas und eines Vorgangsobjektschemas (vgl. Abschnitt 2.3). Vorgangsobjekte und konzeptuelle Objekte tauschen über Kommunikationskanäle Nachrichten aus. Konzeptuelle Objekttypen besitzen einen internen Speicher; sie realisieren mit Hilfe ihrer Methoden die Aktionen eines Vorgangs, dessen Steuerung von einem Vorgangsobjekttyp durchgeführt wird [FeSi98, 223]. Für die Realisierung objektintegrierter Anwendungssysteme sind Plattformen notwendig, die eine transparente Kommunikation von Objekten unterstützen. Dazu zählen z.B. CORBA-konforme Produkte.

## **3.4 Orthogonalität der Rahmenbedingungen**

Zwischen den in den vorangehenden Abschnitten betrachteten Rahmenbedingungen existieren vielfältige Interdependenzen (vgl. z.B. [Sydow85]). Bei der Gestaltung von Aufgabensystem, Aufbauorganisation und dem System der maschinellen Aufgabenträger kann die Berücksichtigung dieser Interdependenzen dazu beitragen, im Sinne der Anforderungen ungeeignete Kombinationen von Ausprägungen der genannten Merkmale zu erkennen.

In diesem Artikel steht jedoch die Gestaltung von Anwendungssystemen im Vordergrund. Aufgabensystem, Aufbauorganisation und maschinelle Aufgabenträger werden deshalb als gegeben angenommen. Um den möglichen Lösungsraum bei der

---

<sup>31</sup> Inwieweit die beiden Integrationsformen die genannten Einzelziele erreichen, wird in [Fers92] näher beschrieben.

Gestaltung des Anwendungssystems nicht einzuschränken, werden keine Restriktionen für die Kombinierbarkeit der beschriebenen Rahmenbedingungen berücksichtigt.

## **4 Entwurfsmuster für verteilte AWS-Architekturen**

In einer hochdynamischen Umwelt stehen Unternehmen unter zunehmendem Druck, sich an neue Rahmenbedingungen anzupassen. Dies gilt vor allem für betriebliche Anwendungssysteme als Aufgabenträger für die Unterstützung von Geschäftsprozessen. Wiederverwendung im Rahmen der Entwicklung von betrieblichen Anwendungssystemen wird schon lange als ein Ansatz angesehen, sowohl die Produktivität als auch die Qualität der Entwicklungsergebnisse zu verbessern [Endr88].

### **4.1 Das Konzept der Patterns für die Wiederverwendung von Modellierungswissen**

Seit Cunningham und Beck 1987 erstmals das Pattern-Konzept aus dem ursprünglichen Bereich der Städte- und Gebäude-Architektur [Alex79] auf das Software-Design übertragen haben [App197], werden Patterns zunehmend als ein geeignetes Mittel angesehen, immaterielle Entwürfe wiederverwendbar zu machen. Liegt der Schwerpunkt bisher jedoch auf software-technischen Patterns (vgl. z.B. [Gamm+95]), so soll im folgenden versucht werden, diesen Ansatz auch für die fachliche Spezifikation von AwS nutzbar zu machen. Mit einem solchen Konzept können Entwurfsentscheidungen wiederverwendbar gemacht werden, die auch Rahmenbedingungen berücksichtigen, wie sie in Abschnitt 3 beschrieben sind.

Von den unterschiedlichen Interpretationen des ursprünglichen Patternbegriffs [Quib96]<sup>32</sup> wird im folgenden der Begriff des generischen Entwurfsmusters genutzt. Ein generisches Entwurfsmuster besteht aus einer kontextsensitiven Beschreibung von Entwurfsobjekt und Entwurfszielen sowie einem generischen Entwurfsverfahren, welches unter Berücksichtigung der Ziele und des jeweiligen Kontextes für eine Anwendung parametrisiert wird. Generische Entwurfsmuster stellen die intersubjektive Nachvollziehbarkeit der generierten Lösung sicher und besitzen damit ein hohes Wiederverwendungspotential [HSW98].

---

<sup>32</sup> Eine vergleichende Darstellung findet sich auch in [HSW98].

## 4.2 Das Konzept des generischen Anwendungssystemmodells

Die Spezifikation von Anwendungssystemen stellt ein schlecht-strukturiertes, komplexes Entwurfsproblem dar. Der Ansatz, der auch dem generischen Geschäftsprozeßmodell [HSW98] zugrundeliegt, leistet einen Beitrag zur Lösung solcher Entwurfsprobleme. Dabei wird die Ausgangssituation durch die Angabe des Entwurfsobjekts und des Kontextes in einem Initialmodell beschrieben. Gleichzeitig wird das ursprünglich komplexe Entwurfsproblem in Teilprobleme zerlegt, von denen jedes einzelne sich leichter lösen läßt als das Gesamtproblem [Simon77]. Die Beschreibung der Teilprobleme und ihrer Lösungsverfahren erfolgt in Form generischer Entwurfsmuster. Um eine integrierte Gesamtlösung zu ermöglichen, werden Beziehungen zwischen den Entwurfsmustern angegeben, so daß ein Pattern-System entsteht. Außerdem können Einzelfallösungen in Form nicht-generischer Module beschrieben werden.

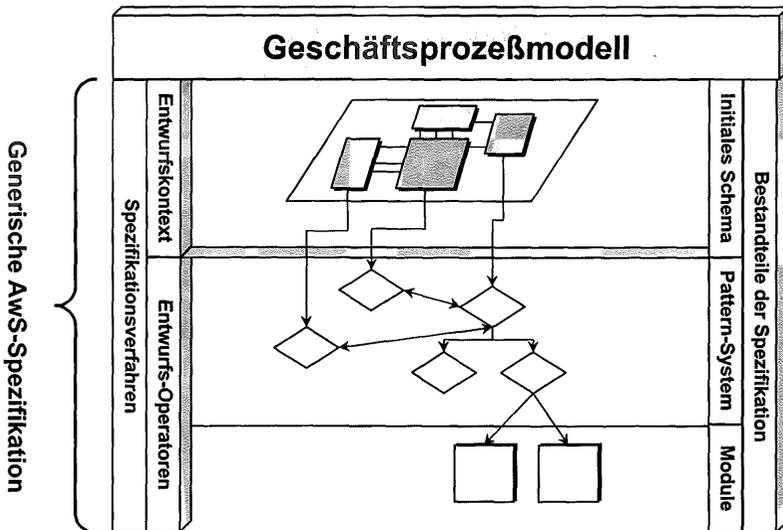


Abbildung 4: Struktur einer generischen Anwendungssystemspezifikation

Überträgt man diesen Ansatz zur Lösung von komplexen Entwurfsproblemen auf das in Abschnitt 2.3 beschriebene Konzept für die fachliche Spezifikation verteilter Anwendungssysteme, so erhält man die Struktur der generischen Anwendungssystemspezifikation (vgl. Abbildung 4). Das initiale Schema beschreibt die aus dem korrespondierenden Geschäftsprozeßmodell abgeleiteten Application Objects (vgl. Abschnitt 2.3) sowie deren Beziehungen als das komplexe Entwurfsobjekt. Das

Pattern-System beschreibt die generischen Entwurfsmuster und deren Beziehungen. Die Beschreibung der Entwurfsmuster folgt dem in [HSW98] angegebenen Beschreibungsrahmen generischer Entwurfsmuster für die Konstruktion von Geschäftsprozeßmodellen<sup>33</sup>. Wichtigste Bestandteile sind der Kontext, die Entwurfsziele, bestehend aus Entwurfssachziel und Restriktionen, sowie das generische Entwurfsverfahren. Letzteres wird im wesentlichen unter Verwendung der Konstruktionsregeln für die fachliche Spezifikation von Anwendungssystemen gemäß der SOM-Methodik [FeSi98, 202ff] formuliert. Darüber hinaus können fachliche Spezifikationen von konkreten Anwendungssystem-Komponenten als Module angegeben werden.

### 4.3 Ein Beispiel für eine generische Anwendungssystemspezifikation

Der bisher entwickelte Ansatz soll nun konkretisiert werden. Zu diesem Zweck wird für den in Abschnitt 2.2 beschriebenen Geschäftsprozeß „Kapitalbeschaffung“ eine generische Anwendungssystemspezifikation entwickelt. Die gemäß der in Abschnitt 2.3 beschriebenen Ableitungsvorschrift entwickelten Application Objects und deren Beziehungen (vgl. Abbildung 3) stellen das initiale Schema für die generische Anwendungssystemspezifikation dar. Auf dieser Basis werden im folgenden beispielhaft zwei Entwurfsmuster in verkürzter Form beschrieben.

#### Entwurfsmuster „Datenintegrierte Refinanzierung“

Bestandteile des Entwurfskontextes sind die Application Objects *Kapitalmarktpartner*, *Refinanzierungsabwicklung* und *-entscheidung* und *Abwicklungssystem* (vgl. Abbildung 3). Organisatorische Restriktionen sind hohe Entscheidungsdezentralisation sowie eine verrichtungsorientierte Arbeitsteilung. Die bestehende Plattform, ein Großrechnersystem mit relationalem Datenbanksystem, soll weiterhin genutzt werden. Das Entwurfsverfahren sieht vor, die Application Objects *Refinanzierungsentscheidung* und *-abwicklung* zu aggregieren. Die in den anderen Application Objects zusammengefaßten Aufgaben sollen nicht durch dieses Teil-Anwendungssystem unterstützt werden. Für das aggregierte Application Object *Refinanzierung* wird die Datenintegration als Realisierungsform gewählt. Dazu werden die konzeptuelle Objekttypen des Application Objects zu einem Datenbankschema zusammengefaßt. Die Vorgangsobjekttypen werden als Funktionen interpretiert, die auf diesem Schema operieren.

---

<sup>33</sup> Basis des Beschreibungsrahmens sind die in [BuMe95] und [Gamm+95] vorgeschlagenen Formen der Beschreibung von Patterns.

### Entwurfsmuster „Integration des Kapitalmarktpartners“

Auch der Kontext dieses Patterns umfaßt die vier Application Objects aus Abbildung 3. Das zu spezifizierende Anwendungssystem soll die zwischenbetriebliche Integration der *Refinanzierungsabwicklung* mit einem ausgewählten *Kapitalmarktpartner* unterstützen. Die Entscheidungsbefugnis bezüglich der Refinanzierungsform verbleibt bei der *Refinanzierungsentscheidung*. Um eine hohe Anpaßbarkeit der Vertriebswege zu ermöglichen, soll das *Abwicklungssystem* mit den Application Objects *Refinanzierungsabwicklung* und *Refinanzierungsentscheidung* lose gekoppelt werden. Das Entwurfsverfahren sieht vor, daß in diesem Teil-Anwendungssystem nur die in den Application Objects *Kapitalmarktpartner* und *Refinanzierungsabwicklung* enthaltenen Aufgaben automatisiert werden. Die Objektintegration dieser Komponenten wird durch eine geeignete Plattform realisiert. Eine CORBA-konforme Middleware ermöglicht beispielsweise die Verwaltung redundanter Schnittstellenobjekte sowie die Konsistenzsicherung der beteiligten Komponenten.

## 5 Fazit und Ausblick

Basierend auf einer Architektur für die Entwicklung verteilter Anwendungssysteme wurde ein Ansatz vorgestellt, der die Einbeziehung technischer und organisatorischer Rahmenbedingungen der Systementwicklung gestattet. Zur Beschreibung von Spezifikationsschritten, die die Verknüpfung spezifischer Rahmenbedingungen und ihren Einfluß auf verteilte Anwendungssysteme berücksichtigen, wurden generische Entwurfsmuster vorgeschlagen. Damit sind folgende Vorteile verbunden:

- Erprobtes Entwurfswissen wird in wiederverwendbarer Form zur Verfügung gestellt.
- Das komplexe Entwurfsproblem der Systementwicklung wird in handhabbare Teilprobleme und zugehörige Lösungsverfahren zerlegt.
- Lösungsverfahren werden prozeßorientiert beschrieben, um ihre Anwendbarkeit zu fördern.
- Generische Entwurfsmuster können auch zur Dokumentation von Entwürfen verwendet werden, um die Nachvollziehbarkeit der Spezifikationen zu erhöhen.

Das vorgestellte Konzept wird aktuell im Verbundprojekt WEGA<sup>34</sup> evaluiert und weiterentwickelt. Insbesondere sollen generische Anwendungssystemspezifikationen für unterschiedliche Domänen erstellt werden.

---

<sup>34</sup> WEGA (Wiederverwendbare und erweiterbare Geschäftsprozeß- und Anwendungssystem-Architekturen) ist ein vom BMBF gefördertes Verbundprojekt, an dem die Universität Bamberg (Lehrstühle Prof. Ferstl und Prof. Sinz), die SAP AG Walldorf und die KPMG Unternehmensberatung GmbH Ffm. beteiligt sind (vgl. [FeSi+98], [FeHa+96]).

## 6 Referenzen

- [Alex79] Alexander C.: The timeless way of building. New York 1979
- [App197] Appleton B.: Patterns and Software: Essential Concepts and terminology. [Http://www.enteract.com/~bradapp/docs/patterns-intro.html](http://www.enteract.com/~bradapp/docs/patterns-intro.html). Stand: 29.07.1998
- [Bull+97] Bullinger H.J., Ilg R., Zinser S.: Turbulente Zeiten erfordern kreative Köpfe – Neue Impulse für das Mangement von Unternehmen. In: Scheer A.-W. (Hrsg.): Organisationsstrukturen und Informationssysteme auf dem Prüfstand. Heidelberg 1997, S. 33-50
- [BuMe95] Buschmann F., Meunier R.: A System of Patterns. In: Coplien J.O., Schmidt D.C. (Eds.): Pattern Languages of Program Design. Reading, MA 1995
- [DaMa93] Davidow W.H., Malone M.S.: Das virtuelle Unternehmen. Frankfurt 1993
- [Diet94] Dietsch H.: Physische Eigenschaften verteilter Rechensysteme: Beiwerk oder Basis? In: Wedekind, H. (Hrsg.): Verteilte Systeme: Grundlagen und zukünftige Entwicklung aus Sicht des Sonderforschungsbereichs 182 „Multiprozessor- und Netzwerkkonfiguration“. Mannheim, Leipzig, Wien, Zürich: BI-Wiss.-Verl., 1994, S. 3 – 16
- [Endr88] Endres A.: Software-Wiederverwendung: Ziele, Wege, Erfahrungen. In: Informatik-Spektrum, Band 11 (1992), S.85-95
- [Ensl78] Esnlow P.H.: What is a „Distributed“ Data Processing System? In: McEntire P.L., O'Reilly J.G., Larson R.E. (Ed.): Distributed Computing: Concepts and Implementations. IEEE Press. New York, 1984, pp 16-23, Reprinted from IEEE Computer vol. 11, pp 13-21, Jan 1978
- [FeHa+96] Ferstl O.K., Hammel C., Keller G., Pfister A., Popp K., Schlitt M., Sinz E.J., Wolf St., Zencke P.: Verbundprojekt WEGA – Wiederverwendbare und erweiterbare Geschäftsprozeß- und Anwendungssystem-Architekturen. In: Statusband des BMBF Softwaretechnologie. Berlin 1996, S. 3-21
- [Fers92] Ferstl, O.K.: Integrationskonzepte Betrieblicher Anwendungssysteme. Fachberichte Informatik der Universität Koblenz-Landau Nr. 1/1992
- [FeSi+97] Ferstl, O.K., Sinz E.J., Hammel C., Schlitt M., Wolf St.: Bausteine komponentenbasierter Anwendungssysteme. In: HMD Theorie und Praxis der Wirtschaftsinformatik 197, September 1997, S.24-46
- [FeSi+98] Ferstl O.K., Sinz E.J., Hammel C., Schlitt M., Wolf St., Popp K., Pfister A.: Verbundprojekt WEGA – Wiederverwendbare und erweiterbare Geschäftsprozeß- und Anwendungssystem-Architekturen. Erscheint in: Statusband des BMBF Softwaretechnologie. Berlin 1998
- [FeSi98] Ferstl, O.K., Sinz E.J.: Grundlagen der Wirtschaftsinformatik, 3. vollst. überarb. und erw. Aufl. München 1998
- [Gamm+95] Gamma E., Helm R., Johnson R., Vlissides J.: Design Patterns. Elements of Reusable Object-Oriented Software. Reading, Massachusetts 1995
- [Gro82] Grochla E.: Grundlagen der organisatorischen Gestaltung. Stuttgart 1982

- 
- [HSW98] Hammel C., Schlitt M., Wolf St.: Pattern-basierte Konstruktion von Unternehmensmodellen. In: Informationssystem-Architekturen, Rundbrief des GI-Fachausschusses 5.2, Heft 1/98, S. 22-37
- [Krum97] Krumbiegel J.: Integrale Gestaltung von Geschäftsprozessen und Anwendungssystemen in Dienstleistungsbetrieben. Wiesbaden 1997
- [Lori88] Lorin H.: Aspects of distributed computer systems. 2<sup>nd</sup> Edition. New York, Wiley 1988
- [Mert93] Mertens P.: Integrierte Informationsverarbeitung Bd.1. 9. vollst. überarb. Auflage, Wiesbaden 1993
- [Quib96] Quibeldey-Cirkel, K.: Das aktuelle Schlagwort: Entwurfsmuster. In: Informatik-Spektrum, Band 19, Heft 6/1996, S. 326-327
- [Scheer91] Scheer, A.-W.: Architektur integrierter Informationssysteme. Berlin 1991
- [Simon77] Simon H.: The Structure of Ill-Structured Problems. In: Simon H.: Models of Discovery. Boston 1977, pp. 304-325. Original in: AI 4, 1973, pp. 181-201
- [Sydow85] Sydow J.: Der soziotechnische Ansatz der Arbeits- und Organisationsgestaltung. Frankfurt 1985
- [Wolf97] Wolf, St.: Application Objects als fachliche Bausteine verteilter Systeme. In: Informationssystem-Architekturen, Rundbrief des GI-Fachausschusses 5.2, Heft 2/97, S. 51-55
- [Zimm97] Zimmermann F.O.: Structural and managerial of virtual Enterprises. Published in: Proceedings of the European Conference on Virtual Enterprises and Networked Solutions – New Perspectives on Management, Communication and Information Technology, April 07th to 10<sup>th</sup> 1997, Paderborn, Germany  
URL: <http://www.seda.sowi.uni-bamberg.de/~fzimmerm/index.html>  
(Stand: 28.07.98)