

Magdeburger Schriften zur Wirtschaftsinformatik

**Klaus Turowski**

## **Fachkomponenten**

Komponentenbasierte betriebliche Anwendungssysteme

Shaker Verlag  
Aachen 2003

Die Deutsche Bibliothek - CIP-Einheitsaufnahme

*Turowski, Klaus:*

Fachkomponenten : Komponentenbasierte betriebliche  
Anwendungssysteme / Klaus Turowski.

Aachen : Shaker, 2003

(Magdeburger Schriften zur Wirtschaftsinformatik)

ISBN3-8322-1359-7

Copyright Shaker Verlag 2003

Alle Rechte, auch das des auszugsweisen Nachdruckes, der auszugsweisen  
oder vollständigen Wiedergabe, der Speicherung in Datenverarbeitungs-  
anlagen und der Übersetzung, vorbehalten.

Printed in Germany.

ISBN 3-8322-1359-7

ISSN 1618-2308

Shaker Verlag GmbH • Postfach 101818 • 52018 Aachen

Telefon: 02407 / 95 96 - 0 • Telefax: 02407 / 95 96 - 9

Internet: [www.shaker.de](http://www.shaker.de) • eMail: [info@shaker.de](mailto:info@shaker.de)

## Geleitwort

---

In diesem Buch geht es um den lange gehegten Traum des Software Engineering: Software soll nicht mehr durch die mühselige, riskante und kostenintensive grundständige Entwicklung „from scratch“, sondern durch das einfache Zusammenstecken vorgefertigter und genormter Komponenten entstehen. Schaut man über den Tellerrand der (Wirtschafts-) Informatik hinweg, dann ist dies wahrlich nichts Neues, sind Komponenten etwa in den Ingenieurwissenschaften seit Jahrzehnten dank nationaler und internationaler Normungsbemühungen Gang und Gäbe. Es stellt sich die ernsthafte Frage, warum etwas, was bei den Ingenieuren offensichtlich schon lange reibungslos funktioniert, in der Informatik trotz mehr als 30 Jahren Forschung im Software Engineering partout nicht gehen soll. Hat die Informatik versagt? Sind Ingenieure vielleicht die besseren Informatiker?

Die Arbeit von Klaus Turowski beantwortet die erste Frage mit einem klaren „Nein“ und lässt die Antwort auf die zweite Frage zumindest offen, zeigen sich bei näherem Hinsehen doch die akademischen Wurzeln des Verfassers im Wirtschaftsingenieurwesen. Das klare „Nein“ als Antwort auf die erste Frage ergibt sich zum Einen aus der Tatsache, dass Software im Vergleich zu materiellen Produkten eben ein besonderer Stoff ist, und zum Anderen daraus, dass die Gestaltung der Komponenten nicht nur eine technische, sondern auch eine fachliche Dimension hat. Speziell in der betriebswirtschaftlichen Fachdomäne geht der Trend bislang eher zu einer terminologischen Diversifizierung (man könnte auch von einer Babylonisierung sprechen) als zur Schaffung einer klaren und vereinheitlichten Fachsprache als Grundlage für die Spezifikation von Fachkomponenten; da kann man nur neidvoll zur Medizinischen Informatik herüberblicken.

Es wäre schon ein wissenschaftshistorischer Treppenwitz, wenn die Betriebswirtschaftslehre die hehre Aufgabe der Standardisierung ihrer Terminologie den Herstellern von ERP-Software oder Wirtschaftsinformatikern überlässt, aber genau diese Tendenz ist heute beobachtbar – es trinkt die Kuh vom Kalb. Es stellt sich aber auch die Frage, ob eine solche Standardisierung mit einem Anspruch auf Vollständigkeit überhaupt sinnvoll in endlicher Zeit bewältigbar ist. Turowski begegnet dieser Problematik mit einem viel versprechenden Ansatz, indem er aufzeigt, bis wohin eine Standardisierung notwendig ist und ab wann Individualkomponenten sinnvoll eingesetzt werden können. Daraus resultiert eine Architektur, die sowohl die notwendigen Restriktionen als auch Freiräume für eine komponentenbasierte Systementwicklung in der betriebswirtschaftlichen Domäne berücksichtigt. Damit hat er einen

notwendigen Grundstein für eine Konstruktionslehre gelegt, die dem oben zitierten Traum des Software Engineering ein beträchtliches Stück näher kommt.

Es wird sich zeigen, ob diese Grundlagen auch die verdiente Akzeptanz bei den Softwareentwicklern findet. Dabei sind weniger fachliche als psychologische Effekte zu fürchten: bis jetzt nutzen Softwareentwickler eher die Zahnbüste als den Code eines anderen Entwicklers. Aber das ist ein anderes Thema...

Magdeburg im Dezember 2002

*Claus Rautenstrauch*

## Vorwort

---

Die Vorteilhaftigkeit der Idee, betriebliche Anwendungssoftware aus auf einem Markt gehandelten Softwarekomponenten zusammensetzen – in Analogie zur Standard-Vorgehensweise in den etablierten Ingenieurwissenschaften, wo z. B. ein Automobilhersteller entscheiden kann, welchen Ölfilter welchen Herstellers er in ein konkretes Fahrzeug als *Komponente* einsetzt – scheint unmittelbar auf der Hand zu liegen. So realisiert man auf der Basis von Plattformkonzepten und Komponentenbauweise in der „traditionellen“ industriellen Konstruktion und Fertigung unbestritten seit Jahren Qualitäts- und Zeitvorteile.

Nun stellt sich jedoch die Frage, ob sich dieses Erfolgsmodell so einfach auf betriebliche Anwendungssoftware übertragen lässt und ob die Software, wie häufig in Werbebroschüren ausgeführt, nicht sowieso schon komponentenorientiert umgesetzt ist. Um diese Fragen beantworten zu können, muss zunächst geklärt werden, was unter einer Softwarekomponente, und insbesondere unter einer fachlichen Softwarekomponente oder kurz Fachkomponente, zu verstehen sein soll. Zieht man für diese Definition ein intuitives Komponentenverständnis heran (die Realisierung einer Komponente bleibt deren Verwender gegenüber verborgen (Black-Box-Prinzip), Komponenten lassen sich mit geringem Aufwand zu neuen Konfigurationen zusammensetzen und Komponenten sind in Konfigurationen einsetzbar, die deren Hersteller nicht geplant haben müssen), dann wird schnell klar, dass Werbebroschüren nicht unbedingt die tatsächliche Softwarearchitektur wiedergeben.

Ausgehend von diesen Einsichten habe ich mich mit dieser Habilitationsschrift der Herausforderung gestellt, das Themengebiet „komponentenbasierte betriebliche Anwendungssysteme“ aus der Sicht der Wirtschaftsinformatik zu fundieren. Im Laufe der Arbeit wurde dabei immer deutlicher, dass dieses Thema – nicht zuletzt aufgrund einer bestenfalls im Ansatz vorhandenen (wirtschafts-)informatischen Konstruktionslehre – auch in Zukunft noch viele Fragen aufwerfen wird und damit eine dankbare Grundlage für ernsthafte Langzeitforschung liefert.

Abschließend möchte ich allen danken, die mich wissenschaftlich gefördert und bei der Entstehung dieser Arbeit unterstützt haben. Allen voran bedanke ich mich bei meiner Familie und bei meinem Habilitationsvater, Herrn Prof. Dr. Claus Rautenstrauch, der mich fachlich und persönlich in besonderer Weise gefördert und mir darüber hinaus ausreichend Freiraum für die eigene wissenschaftliche Entwicklung und Arbeit gelassen hat. Ferner danke ich Herrn Prof. Dr. Erich Ortner sowie Herrn Prof. Dr. Gunter Saake für Ihre wertvollen Anregungen

und die Übernahme der weiteren Habilitationsgutachten. Bei Herrn Moritz Weizmann bedanke ich mich für die druckfähige Aufbereitung des Manuskripts.

Augsburg im Dezember 2002

*Klaus Turowski*

## Inhaltsverzeichnis

---

Geleitwort .....	iii
Vorwort .....	v
Inhaltsverzeichnis .....	vii
Abkürzungsverzeichnis .....	xi
Symbolverzeichnis .....	xiii
Abbildungsverzeichnis .....	xv
Tabellenverzeichnis .....	xix
<b>1 Einleitung .....</b>	<b>1</b>
1.1 Komponentenorientierung als Architekturparadigma .....	1
1.2 Thema, Einordnung und Aufbau .....	5
1.2.1 Thema und Einordnung .....	5
1.2.2 Aufbau .....	6
1.3 Leitbild und Anwendungsszenarien .....	9
1.4 Komponenten und Fachkomponenten .....	15
1.4.1 Komponentenbegriff .....	15
1.4.2 Formale Fassung des Komponentenbegriffs .....	19
1.5 Ordnungsrahmen für komponentenbasierte Anwendungssysteme .....	28
1.5.1 Vom Monolithen zum komponentenbasierten Anwendungssystem .....	28
1.5.2 BCArch: Generelle Architektur komponentenbasierter Anwendungssysteme .....	36
1.5.3 BCLifeCycle: (Produkt-)Lebenszyklus einer Fachkomponente .....	43
1.5.4 CoBCoM: Ordnungsrahmen für komponentenbasierte betriebliche Anwendungssysteme .....	50
<b>2 Standardisierung von Fachkomponenten .....</b>	<b>53</b>
2.1 Formale Spezifikation und Objekte der Standardisierung .....	53
2.1.1 Syntaxebene .....	58
2.1.2 Verhaltensebene .....	61
2.1.3 Intra-Komponenten-Abstimmungsebene .....	66
2.1.4 Inter-Komponenten-Abstimmungsebene .....	69
2.1.5 Qualitätsebene .....	74
2.1.6 Domänenebene .....	76
2.2 Einflussfaktoren auf die Standardisierung von Fachkomponenten .....	79
2.2.1 Komponentenmärkte, fachliche Konflikte und Granularität der standardisierten Fachkomponenten .....	80
2.2.2 Berücksichtigung bestehender Standards und Integration von Alt-Anwendungen .....	89
2.2.3 Technische Aspekte der Standardisierung von Fachkomponenten .....	93
2.3 Klassifikation von Bestrebungen zur Standardisierung von Fachkomponenten .....	95
<b>3 Entwicklung .....</b>	<b>99</b>
3.1 Aufgabe und Umfang .....	99
3.2 Vorgehensmodelle .....	100
3.2.1 Klassische und prototypingorientierte Softwareentwicklungsmethodiken .....	102
3.2.2 Objektorientierte Softwareentwicklungsmethodiken .....	108
3.2.3 Komponentenorientierte Ansätze .....	112
3.2.3.1 Twin-Life-Cycle-Modell .....	112
3.2.3.2 SELECT Perspective .....	115
3.2.3.3 Catalysis .....	118
3.2.3.4 WebComposition Vorgehensmodell .....	121
3.3 Aufgabenbezogene Betrachtungen .....	123
3.3.1 Domänenanalyse und Referenzmodellierung .....	123
3.3.2 Spezifikation von Fachkomponenten .....	125
3.3.2.1 Notwendigkeit der Verwendung standardisierter Notationen .....	125
3.3.2.2 Erweiterung der OCL um temporale Operatoren .....	127
<b>4 Anpassung .....</b>	<b>137</b>
4.1 Technische Anpassung .....	137
4.1.1 Entstehung von Integrationshemmnissen .....	137
4.1.2 Arten technischer Inkompatibilitäten .....	140

4.1.3	Behebung technischer Inkompatibilitäten .....	141
4.2	Fachliche Anpassung .....	143
4.2.1	Aufgaben der fachlichen Anpassung .....	143
4.2.2	Techniken der fachlichen Anpassung .....	145
4.2.3	Probleme der fachlichen Anpassung .....	147
<b>5</b>	<b>Komposition.....</b>	<b>151</b>
5.1	Aufgaben der Komposition .....	151
5.2	Kompositionstechniken .....	155
5.2.1	Techniken zur Realisierung von Kommunikationskanälen .....	155
5.2.1.1	Softwarebusse .....	156
5.2.1.2	Ereigniskanäle .....	157
5.2.1.3	Tupelräume .....	160
5.2.2	Kompositionstechniken zur Realisierung der zeitlichen Abstimmung.....	163
5.2.2.1	Skriptsprachen .....	163
5.2.2.2	Workflowmanagementsysteme.....	165
5.2.3	Behebung fachlicher Konflikte im Rahmen der inhaltlichen Abstimmung mit Linkobjekten .....	168
5.2.4	Techniken zur strukturellen Komposition .....	174
<b>6</b>	<b>Evolution und Deinstallation .....</b>	<b>181</b>
6.1	Evolution .....	181
6.1.1	Aufgaben und Definition .....	181
6.1.2	Austausch von Fachkomponenten .....	183
6.1.3	Beibehaltung des Systemzustands bei Versionswechseln .....	190
6.1.4	Erweiterbarkeit von Fachkomponenten .....	192
6.1.4.1	Notwendigkeit der Unterstützung von Erweiterbarkeit .....	192
6.1.4.2	XML zur Realisierung unabhängiger Erweiterbarkeit .....	193
6.1.4.3	Beispiele zur XML-basierten Erweiterung komponentenbasierter Anwendungssysteme .....	196
6.2	Deinstallation .....	200
<b>7</b>	<b>Komponenten-Anwendungs-Frameworks .....</b>	<b>203</b>
7.1	Klassifizierung von Komponenten-Frameworks .....	203
7.2	San Francisco Framework .....	207
7.2.1	Motivation .....	207
7.2.2	Architektur .....	208
7.2.3	Klassifikation .....	212
7.3	SAP Business Framework .....	214
7.3.1	Motivation .....	214
7.3.2	Architektur .....	215
7.3.3	Klassifikation .....	220
7.4	Object Management Architecture: Domain Interfaces .....	222
7.4.1	Motivation .....	222
7.4.2	Architektur .....	222
7.5	Weitere Ansätze .....	226
<b>8</b>	<b>Komponenten-System-Frameworks .....</b>	<b>227</b>
8.1	Object Management Architecture: Common Object Request Broker Architecture .....	227
8.1.1	Architektur .....	227
8.1.2	Anwendungsübergreifende Dienste .....	229
8.1.3	Klassifikation .....	234
8.2	Enterprise JavaBeans .....	236
8.2.1	Motivation .....	236
8.2.2	Architektur .....	237
8.2.3	Anwendungsübergreifende Dienste .....	241
8.2.4	Klassifikation .....	243
8.3	Distributed Component Object Model .....	245
8.3.1	Motivation .....	245
8.3.2	Architektur .....	246
8.3.3	Anwendungsübergreifende Dienste .....	249
8.3.4	Klassifikation .....	252

<b>9 Zusammenfassung und Ausblick.....</b>	<b>255</b>
9.1 Zusammenfassung.....	255
9.2 Ausblick.....	256
<b>A Anhang.....</b>	<b>259</b>
A.1 Grammatik der um temporale Operatoren erweiterten OCL.....	259
<b>Literaturverzeichnis .....</b>	<b>261</b>
<b>Glossar .....</b>	<b>277</b>
<b>Personenregister.....</b>	<b>283</b>
<b>Stichwortverzeichnis .....</b>	<b>287</b>