

LEHRSTUHL FÜR
ALLG. BWL UND WIRTSCHAFTSINFORMATIK
UNIV.-PROF. DR. HERBERT KARGL

Maurer, Gerd

**CORBA-basierte
Workflow-Architekturen**

***Die objektorientierte Kernanwendung
der Bausparkasse Mainz AG***

ARBEITSPAPIERE WI
Nr. 12/1997

Schriftleitung:
Dr. rer. pol. Axel C. Schwickert

Information

- Reihe:** Arbeitspapiere WI
- Herausgeber:** Univ.-Prof. Dr. Axel C. Schwickert
Professur für BWL und Wirtschaftsinformatik
Justus-Liebig-Universität Gießen
Fachbereich Wirtschaftswissenschaften
Licher Straße 70
D – 35394 Gießen
Telefon (0 64 1) 99-22611
Telefax (0 64 1) 99-22619
eMail: Axel.Schwickert@wirtschaft.uni-giessen.de
<http://wi.uni-giessen.de>
- Bis Ende des Jahres 2000 lag die Herausgeberschaft bei:
- Lehrstuhl für Allg. BWL und Wirtschaftsinformatik
Johannes Gutenberg-Universität Mainz
Fachbereich Rechts- und Wirtschaftswissenschaften
Welderweg 9
D - 55099 Mainz
- Ziele:** Die Arbeitspapiere dieser Reihe sollen konsistente Überblicke zu den Grundlagen der Wirtschaftsinformatik geben und sich mit speziellen Themenbereichen tiefergehend befassen. Ziel ist die verständliche Vermittlung theoretischer Grundlagen und deren Transfer in praxisorientiertes Wissen.
- Zielgruppen:** Als Zielgruppen sehen wir Forschende, Lehrende und Lernende in der Disziplin Wirtschaftsinformatik sowie das IuK-Management und Praktiker in Unternehmen.
- Quellen:** Die Arbeitspapiere entstanden aus Forschungsarbeiten, Diplom-, Studien- und Projektarbeiten sowie Begleitmaterialien zu Lehr- und Vortragsveranstaltungen des Lehrstuhls für Allg. Betriebswirtschaftslehre und Wirtschaftsinformatik Univ. Prof. Dr. Herbert Kargl an der Johannes Gutenberg-Universität Mainz.
- Hinweise:** Wir nehmen Ihre Anregungen und Kritik zu den Arbeitspapieren aufmerksam zur Kenntnis und werden uns auf Wunsch mit Ihnen in Verbindung setzen.
Falls Sie selbst ein Arbeitspapier in der Reihe veröffentlichen möchten, nehmen Sie bitte mit dem Herausgeber (Gießen) unter obiger Adresse Kontakt auf.
Informationen über die bisher erschienenen Arbeitspapiere dieser Reihe und deren Bezug erhalten Sie auf dem Schlußblatt eines jeden Arbeitspapiers und auf der Web Site des Lehrstuhls unter der Adresse <http://wi.uni-giessen.de>

Arbeitspapiere WI Nr. 12/1997

- Autoren:** Maurer, Gerd
- Titel:** CORBA-basierte Workflow-Architekturen – Die objektorientierte Kernanwendung der Bausparkasse Mainz AG
- Zitation:** Maurer, Gerd: CORBA-basierte Workflow-Architekturen – Die objektorientierte Kernanwendung der Bausparkasse Mainz AG, in: Arbeitspapiere WI, Nr. 12/1997, Hrsg.: Lehrstuhl für Allg. BWL und Wirtschaftsinformatik, Johannes Gutenberg-Universität: Mainz 1997.
- Kurzfassung:** Bei der Initialisierung von Workflow-Projekten zur Unterstützung von Geschäftsprozessen stellt sich die Frage, ob und warum ein Unternehmen angesichts der zahlreichen, auf dem Markt erhältlichen Standardsysteme ein individuelles Workflow-System entwickeln sollte. Eine Reihe von Argumenten deutet darauf hin, daß individuelle Eigenentwicklungen durchaus eine erwägenswerte Alternative zu den existierenden Standardsystemen darstellen. Aus der Diskussion dieses Aspektes ergibt sich u. a. die Frage, ob sich die Eigenentwicklung eines CORBA-konformen Workflow-Systems lohnt. Vielversprechende, bereits realisierte Systeme setzen auf die standardisierte, durchgängig objektorientierte Architektur der Object Management Group (OMG). Deren Standard „CORBA,, (Common Object Request Broker) bietet zukunftsweisende technologische Vorteile (z. B. Verteiltheit, Plattformunabhängigkeit, Interoperabilität, Modularität) und weist Synergieeffekte zum Workflow-Konzept auf. Die Bausparkasse Mainz AG (BKM) hat sich bereits 1996 für die Eigenentwicklung eines CORBA-konformen Workflow-Systems entschieden; die neue Kernanwendung „BKMJoker“ der BKM wird als Abschluß des vorliegenden Beitrages skizziert.
- Schlüsselwörter:** Workflow-Systeme, CORBA, ORB, Eigenentwicklung, Objektorientierung, Workflow-Architektur, BKMJoker

Inhaltsverzeichnis

1	Aktuelle Fragestellungen im Workflow-Bereich.....	3
1.1	Definition von Workflow-Systemen	3
1.2	Kernfragen im Workflow-Bereich	4
2	Eigenentwicklung vs. Standardlösungen	4
3	CORBA-konforme Workflow-Systeme	7
3.1	Zur Relevanz von CORBA als Architekturbasis.....	7
3.2	Der CORBA-Standard.....	7
3.3	CORBA-konforme Workflow-Architekturen	10
3.4	Ein Praxisbeispiel: Die CORBA-basierte Architektur der BKM	15
	Literaturverzeichnis.....	18

1 Aktuelle Fragestellungen im Workflow-Bereich

1.1 Definition von Workflow-Systemen

Die Planung und Umsetzung prozeßorientierter Organisationsstrukturen bedarf einer intensiven Unterstützung durch spezielle Informationssysteme. Entscheidende Impulse erhoffen sich viele Unternehmen in diesem Zusammenhang vor allem von *Workflow-Management-Systemen*. Man versteht darunter Systeme, welche die Vorgangsbearbeitung auf der Basis von formal modellierten Prozeßdefinitionen steuern. Diese sehr allgemeine Definition wird häufig in der Literatur zitiert, muß jedoch konkretisiert werden.

Im Rahmen der *Modellierungskomponente* werden die Definitionen der Prozeßstrukturen in einem graphischen Editor erstellt, der zusätzliche Animations- und Simulationsfunktionen bietet oder eine komfortable Schnittstelle zu speziellen Modellierungssystemen (z. B. ARIS Toolset, Bonapart) aufweist. Dabei sind die Bearbeiter von Vorgängen dem System nicht nur namentlich bekannt, sondern in einem aufbauorganisatorischen Modell hinterlegt, das neben Rollendefinitionen (z. B. „Kreditprüfer“ anstelle „Sachbearbeiter G. Müller“) auch komplexere organisatorische Beziehungen erfassen kann (z. B. Stellvertretungen, Kompetenzen, Verantwortlichkeiten).

Die *Steuerung* der aus einer Prozeßdefinition (z. B. „Antrag auf Darlehensauszahlung“) erzeugten Instanzen (syn. Geschäftsvorfälle, Fälle, wie z. B. „Antrag des Kunden Müller“) beinhaltet idealerweise die folgenden Funktionalitäten (*Steuerungs- oder auch Ausführungskomponente*): Die zur Bearbeitung anstehenden Prozeßschritte werden termingerecht in den Arbeitsvorrat (syn. Tätigkeitsliste, Work Queue, Work List) des korrekten Bearbeiters eingestellt. Wenn der betreffende Mitarbeiter aus seinem Arbeitsvorrat einen Schritt zur Bearbeitung auswählt, ruft das Workflow-System die passende Applikation (z. B. Tabellenkalkulationssystem, R/3-Transaktion) zusammen mit den zum jeweiligen Schritt gehörenden Daten auf. Je nachdem, wie das System technisch konzipiert ist, macht es die Daten verfügbar (z. B. durch Datenbankoperationen) oder stellt sie dem Bearbeiter direkt zu (z. B. Routing über eine Messaging-Komponente).

Die „Ausführung“ der Vorgänge (Erzeugen der Instanzen, Steuerung der Reihenfolgen, Verwaltung der individuellen Arbeitsvorräte, Terminkontrolle usw.) wird durch die sog. *Workflow Engine* übernommen, eine Programmkomponente des Workflow-Systems, welche vor allem die Algorithmen zur Vorgangsabwicklung enthält. Es ist auch möglich, daß bei der Abarbeitung eines einzelnen Vorganges mehrere verteilte Workflow Engines kooperieren.

Darüber hinaus wird die Vorgangsbearbeitung vom Workflow-System protokolliert, so daß laufend der Bearbeitungsstatus abrufbar ist (z. B. „in Bearbeitung beim Schritt Bonitätsprüfung durch Bearbeiter Müller“) und weitere Prozeßinformationen (z. B. Durchlaufzeiten, Prozeßkosten) generiert werden können (*Informationskomponente*). Hier ist weitergehend ein Anschluß an Systeme wie z. B. EIS (Executive Information System), DSS (Decision Support System) und Data Warehouse vorstellbar.

1.2 Kernfragen im Workflow-Bereich

Die groben Steuerungs-/Ausführungsfunktionalitäten lassen sich sehr detailliert in einzelne Funktionalitäten verfeinern (z. B. Ablehnung von Prozeßschritten durch Bearbeiter, Stornierung von Schritten, Anmahnung von überfälligen Tätigkeiten, Ergänzung durch Archivierungsfunktionen). Derzeit existieren zahlreiche Standard-Workflow-Systeme auf dem Markt (z. B. FlowMark, Staffware), die ein umfangreiches Spektrum derartiger Einzelfunktionalitäten bieten und sich in vielerlei Hinsicht voneinander differenzieren. Das potentielle Anwenderunternehmen steht angesichts der Vielfalt der Systeme und der einzelnen Workflow-Funktionalitäten vor der Frage, welches System für seine Unternehmensprozesse geeignet ist, denn nicht jedes System und nicht jede Funktionalität paßt zu jedem Prozeßtypen. Eine fundierte Systemplanung muß eine grundsätzliche Frage zwingend berücksichtigen, die in Literatur und Praxis häufig vernachlässigt wird: die Entscheidung zwischen der Entwicklung einer Individuallösung und der Beschaffung eines Standardsystems, die in Kapitel 2 angesprochen wird. Im Zusammenhang mit diesem Entscheidungsproblem ergibt sich eine weitere grundlegende Fragestellung: Die Workflow-Komponente muß harmonisch in die IT-Gesamtarchitektur eingebettet werden; dabei sind die Art und Weise sowie das Ausmaß der Integration festzulegen. Diese Kernfrage stellt sich vor allem im Hinblick auf eine CORBA-konforme IT-Architektur, von der man sich zahlreiche Synergieeffekte zum Workflow-Konzept verspricht (Kapitel 3). Die erhältlichen Standard-Workflow-Systeme erweisen sich hier als problematisch.

Die aktuelle praktische Relevanz beider Kernfragen verdeutlicht die IT-Architektur der Bausparkasse Mainz AG (BKM). Die neue Kernanwendung (BKMJoker) der BKM ist grundsätzlich CORBA-basiert und enthält eine CORBA-konforme Workflow-Komponente.

2 Eigenentwicklung vs. Standardlösungen

Eigenentwicklungen sind heute im Workflow-Bereich trotz des hohen Entwicklungsaufwandes¹ durchaus eine ernstzunehmende Alternative zu Standardsystemen; besonders in der Praxis wird diese Alternative nur selten in Betracht gezogen. Die Vorteile einer Eigenentwicklung sind nicht zuletzt aus der Problematik von Standardlösungen heraus zu begründen. Einige hochrelevante Kriterien zur Entscheidung zwischen individueller Eigenentwicklung und der Beschaffung von Standardsystemen zeigt Tabelle 1.

1 Vgl. Schwab, Klaus: Koordinationsmodelle und Softwarearchitekturen als Basis für die Auswahl und Spezialisierung von Workflow-Management-Systemen, in: Vossen, G.; Becker, J. (Hrsg.): Geschäftsprozeßmodellierung und Workflow-Management; Bonn; Albany: Internat. Thomson Publ. 1996, S. 316. Diese Aussage muß allerdings gemäß der Komplexität des zu entwickelnden Systems differenziert werden, denn es können auch sehr einfache Systeme entwickelt werden.

Kriterium	Individuelle Eigenentwicklung	Standardsystem
Kosten	Abhängig vom Funktionsumfang, aber i. d. R. hoher Entwicklungsaufwand. Externes Know-how ist relativ teuer.	Hohe Produktkosten, wenn viele Arbeitsplätze ausgerüstet werden sollen und nach der Anzahl der Lizenzen gezahlt wird. Trotzdem i. d. R. weit niedrigere Kosten als bei Eigenentwicklungen.
Funktionalität	Die von den Unternehmensprozessen und der IT-Umgebung her notwendigen Funktionalitäten können direkt entwickelt werden.	Häufig sind hinsichtlich der notwendigen Funktionalitäten Kompromisse notwendig: möglich ist sowohl zu viel als auch zu wenig Funktionalität. Customizing ist kaum möglich oder äußerst aufwendig.
Anpaßbarkeit, Erweiterbarkeit	Es ist möglich, zukünftige Anpassungen von Anfang an zu berücksichtigen (z. B. Skalierbarkeit, Vorgangsabwicklungsalgorithmus, Oberfläche) oder Anpassungen später in eigener Regie durchzuführen.	Die entscheidenden Elemente des Workflow-Systems, insbesondere die Workflow Engine, werden von den Herstellern nicht offengelegt. Das „Nachprogrammieren“ fehlender Funktionen gilt als sehr aufwendig, ² wenn dies überhaupt möglich ist. Wenn Anpassungen nur durch den Hersteller möglich sind, besteht die Gefahr einer starken Herstellerabhängigkeit.
Prototyping	Prototypen und Testinstallationen sind erst sehr spät im Projekt möglich.	Prototypen-, Test- oder Pilotinstallationen sind schon früh möglich.
Modellierungsmethode	Es kann eine für die Unternehmensbedürfnisse geeignete Methode zur Prozeßmodellierung entworfen oder übernommen werden.	Die Methode eines Standardsystems muß zur betriebswirtschaftlichen Prozeßmodellierung im Unternehmen „passen“.
Einbindung in die gesamte IT-Architektur	Eine Eigenentwicklung kann von Anfang an auf die gesamte IT-Architektur abgestimmt werden.	Es besteht die Gefahr, daß das Workflow-System einen heterogenen Faktor in der IT-Architektur darstellt.
Zukunftssicherheit	Das Unternehmen kann selbst bestimmen, in welche Richtung sich das Produkt weiterentwickeln soll.	Der Hersteller bestimmt, in welche Richtung sich das Produkt weiterentwickelt.
Risiko	Das Risiko einer Eigenentwicklung ist hoch anzusetzen, wenn fundiertes Entwicklungs-Know-how nicht dauerhaft verfügbar ist. Überzogene Ansprüche und Anforderungen können zu überkomplexen Systemstrukturen führen.	Über Referenzinstallationen, Testinstallationen und Herstellergewährleistung kann das Risiko reduziert werden. Aber: „Angegebene Referenzen entpuppen sich nur selten als völlig falsch, aber ebenso selten als völlig richtig. Die Stunde der Wahrheit schlägt dann bei der eigenen Testinstallation.“ ³

Tab. 1: Kriterien zur Entscheidung zwischen Eigenentwicklung und Standardlösung im Workflow-Bereich (Beispiele)

2 Vgl. Schwab, Klaus: Koordinationsmodelle und Softwarearchitekturen als Basis für die Auswahl und Spezialisierung von Workflow-Management-Systemen, a. a. O., S. 298.

3 Grell, Rainer: Elektronische Bearbeitung schwach strukturierter Vorgänge, in: Office Management 6/1995, S. 37.

Die Tabelle verdeutlicht, daß Standardsysteme in aller Regel Kompromisse erfordern. Als problematisch erweist sich unter anderem, daß die Markttransparenz trotz zahlreicher Studien⁴ nur begrenzt ist, so daß der Evaluationsprozeß sehr aufwendig werden kann. Zudem ist ein gewisses Restrisiko, daß möglicherweise doch das „falsche“ System beschafft wird, nicht auszuschließen, denn insbesondere bei hochkomplexen, anspruchsvollen Kernfunktionalitäten (z. B. Transaktionsabsicherung, Änderungen der Prozeßdefinition von aktiven Instanzen) erweist sich die Qualität des Workflow-Systems häufig erst im produktiven Betrieb.

Auch individuelle Eigenentwicklungen beinhalten spezielle Risiken, wie Tabelle 1 zeigt. Trotz des damit verbundenen Aufwands und Risikos hat die BKM sich dafür entschieden, ein individuelles System zu entwickeln. Ausschlaggebend dafür waren unter anderem die folgenden Gründe:

- Eine individuelle Anpassung von Standardsystemen ist nur begrenzt möglich. So ist z. B. die Anpassung der Vorgangsabwicklungsalgorithmen der Workflow Engine i. d. R. nicht möglich.
- Die nach außen hin sichtbare und nutzbare Modularität von Standardsystemen ist unzureichend, d. h., einzelne Komponenten (z. B. Transaktionsdienst, Modellierungskomponente für die Aufbauorganisation) lassen sich nicht austauschen.
- Bei der BKM besteht hoher Bedarf an Ablaufsicherheit (Recovery der Instanzen und der zugehörigen Fachdaten im Fehlerfall). Diese Anforderung ist ohne ein ausgereiftes Transaktionskonzept nicht zu erfüllen. Standardsysteme weisen zwar teilweise Transaktionskonzepte auf; diese beschränken sich jedoch auf das Workflow-System selbst und erlauben es nicht, die Persistenz der Instanzen außerhalb des Workflow-Systems aufrechtzuerhalten und die von den aufgerufenen Applikationen ausgelösten Datentransaktionen zu kontrollieren. Eine individuelle Anpassung der Transaktionsbasis von Standardsystemen ist jedoch nicht oder nur mit hohem Aufwand möglich.
- Ein entscheidendes Argument für die Entwicklung einer neuen Lösung war die notwendige Homogenität der Workflow-Komponente mit der gesamten IT-Architektur des Unternehmens. Es bestand die Gefahr, daß zugekaufte Standardsysteme einen monolithischen Block in einer ansonsten hochmodularen IT-Architektur darstellen (dies wird besonders in Anbetracht von CORBA-konformen IT-Architekturen deutlich; vgl. Kapitel 3). Wenn ein zugekauftes Workflow-System standardisierte Schnittstellen aufweist (z. B. gemäß den Standards der Workflow Management Coalition), werden häufig Funktionen realisiert, die in den Komponenten der umgebenden IT-Architektur schon vorhanden sind (funktionale Redundanzen, z. B. in Form eines für das Workflow-System spezifischen Transaktionskonzeptes); die modulare Austauschbarkeit einzelner Komponenten des Workflow-Systems (z. B. Engine,

4 Z. B. Götzer, Klaus: Workflow - Unternehmenserfolg durch effizientere Arbeitsabläufe; München: Computerwoche-Verl. 1995, Koch, Olaf G.; Zielke, Frank: Workflow Management - Prozeßorientiertes Arbeiten mit der Unternehmens-DV; Haar bei München: Markt und Technik, Buch- und Software-Verl. 1996, Erdl, Günter; Schönecker, Horst G.: Studie: Geschäftsprozeßmanagement - Vorgangssteuerungssysteme und integrierte Vorgangsbearbeitung; München: FBO - Fachverlag für Büro und Organisationstechnik GmbH 1992.

Transaktionskomponente) ist nur sehr bedingt möglich. Die BKM hat sich daher auf eine CORBA-konforme IT-Architektur mit einer CORBA-konformen Workflow-Komponente festgelegt (vgl. Kapitel 3).

3 CORBA-konforme Workflow-Systeme

3.1 Zur Relevanz von CORBA als Architekturbasis

Damit die Workflow-Unterstützung ihre steuernde Rolle in der IT-Architektur spielen kann, ist die Herstellung einer grundsätzlichen „Harmonie“ zwischen dem Workflow-System und allen anderen betroffenen Komponenten der vorhandenen IT-Architektur zwingend erforderlich. Das Workflow-System darf keinen isolierten, monolithischen Block in einer ansonsten hochmodularen IT-Architektur darstellen; alle Workflow-Komponenten müssen flexibel nutzbar, anpaßbar und austauschbar sein. Aufwendige, individuelle Schnittstellen oder funktionale Redundanzen erhöhen die Komplexität und bringen dadurch einen höheren Pflegeaufwand sowie Fehleranfälligkeit mit sich.

In diesem Zusammenhang wird empfohlen, die gesamte IT-Architektur konform zum CORBA-Standard zu gestalten. Grund dafür sind die Vorteile von CORBA: Nutzung der Mechanismen der Objektorientierung (z. B. Vererbung, Klassenbildung, Kapselung von Daten und Funktionen), Verteiltheit, Plattformenunabhängigkeit, Vermeidung von Redundanzen durch die Nutzung modularer Dienste. Aufgrund der Grundstruktur des CORBA-Standards bestehen zudem einige Synergien zum Workflow-Konzept, die von den aktuellen Standardsystemen nicht oder nicht ausreichend genutzt werden können.

3.2 Der CORBA-Standard

CORBA (Common Object Request Broker Architecture) ist ein Architekturstandard der OMG (Object Management Group),⁵ der die Kommunikation und Kooperation verteilter Systeme mit Hilfsmitteln der Objektorientierung und auf der Basis des Client/Server-Prinzips ermöglichen soll.⁶ Systeme, deren Architekturen gemäß dem CORBA-Standard aufgebaut sind, können einen äußerst hohen Grad an Interoperabilität erreichen, d. h., die Kommunikation und Kooperation der Anwendungen wird einfach, standardisiert,

5 Die OMG ist eine Interessensgemeinschaft von zur Zeit ca. 700 Unternehmen, deren Ziel vor allem die Verbreitung und Standardisierung objektorientierter Techniken ist; vgl. Jablonski, Stefan; Böhm, Markus; Schulze, Wolfgang (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, Heidelberg: dpunkt-Verl. 1997, S. 276.

6 Vgl. Georgakopoulos, Dimitrios; Hornick, Mark; Sheth, Amit: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, in: Distributed and Parallel Databases 3/1995, S. 141, Czap, Hans: Common Object Request Broker (CORBA) Architecture, in: Information Management 1/1996, S. 71.

wenig aufwendig und sicher realisiert, indem eine kontrollierte Interaktion zwischen verteilten Objekten hergestellt wird.⁷

Objekte (Application Objects, Business Objects) kapseln die fachlich relevanten Daten und Funktionen.⁸ Zum Beispiel enthält das Geschäftsobjekt „Rechnung“ sowohl die Daten einer Rechnung als auch deren Funktionen („Methoden“), wie das Drucken und das Prüfen der Rechnung. Die Interaktion zwischen Client- und Server-Objekten und allgemeinen Diensten der CORBA-Architektur wird über spezielle Programme, sog. ORBs (Object Request Broker), gesteuert. Vereinfacht dargestellt sucht der ORB zu jeder Anfrage (Request) eines Client-Objektes das korrekte Ziel-Objekt (Implementation Object) heraus und stellt dessen Methoden und Daten zur Verfügung.⁹ Der Vorteil dieser Technik liegt darin, daß die Client-Anwendungen (z. B. Workflow-Systeme) nicht die Lokation, Datenrepräsentation und Implementierungssprache der Implementation Objects kennen müssen – dies wird vom ORB übernommen. Der ORB ist somit quasi ein „Postverteiler“.¹⁰ Die Schnittstellen zu den Objekten (Aufruf, Datenübergabe usw.) werden in einer standardisierten Sprache, der Interface Definition Language (IDL), formuliert.¹¹ IDL-Programme fungieren praktisch als Adapter.

Abb. 1 stellt die Funktionsweise der Client/Server-Kommunikation über ORB und IDL schematisch dar.

Über die objektspezifischen Methoden hinaus werden weitere, allgemein von Objekten verwendete Dienste angeboten:¹²

- *Object Services*: Jedes Objekt muß bestimmte grundsätzliche Dienste bereitstellen, z. B. das Erzeugen neuer Objekte und die Benachrichtigung über eingetretene Ereignisse. Darüber hinaus werden bestimmte infrastrukturelle Dienste nicht von allen Objekten benötigt, so z. B. *Transaction Services*, die sicherstellen sollen, daß die durchgeführten Transaktionen (Operationen auf Objekte) aufgezeichnet und kontrolliert werden.

7 Vgl. Georgakopoulos, Dimitrios et al.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, a. a. O., S. 141.

8 Zum Business-Object-Konzept vgl. z. B. Casanave, Cory: Business-Object Architectures and Standards, URL: <http://www.dataaccess.com/bodtf/BOPaper.htm> (Stand: 12.5.97), Miami, Florida 1997, passim.

9 Auf dem Markt existiert schon eine Reihe von konkreten, erprobten ORB-Produkten; vgl. Georgakopoulos, Dimitrios et al.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, a. a. O., S. 142, Rösch, Martin: OMG-Standards und ihre Bedeutung für die Praxis, in: OBJEKTSpektrum 1/1994, S. 21, Miller, John A.; Planiswami, Devanand; Sheth, Amit P.; Kochut, Krys J.; Singh, Harvinder: WebWork: METEOR²'s Web-based Workflow Management System, Technical Report #UGA-CS-TR-97-002, Department of Computer Science, University of Georgia, März 1997, S. 5.

10 Vgl. Rösch, Martin: OMG-Standards und ihre Bedeutung für die Praxis, a. a. O., S. 19.

11 Vgl. z. B. Rösch, Martin: OMG-Standards und ihre Bedeutung für die Praxis, a. a. O., S. 20.

12 Vgl. Georgakopoulos, Dimitrios et al.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, a. a. O., S. 142, Rösch, Martin: OMG-Standards und ihre Bedeutung für die Praxis, a. a. O., S. 19, Yang, Zhonghua; Duddy, Keith: CORBA: A Platform for Distributed Object Computing (A State-of-the-Art Report on OMG/CORBA), a.a.O., S. 5.

- *Common Facilities*: Diese Dienste werden zwar häufig, aber nicht von allen Objekten benötigt, z. B. Druckausgabe und Wiedervorlage. Im Gegensatz zu den Object Services sind die Dienste der Common Facilities anwendungsnäher.
- Die *Domain Interfaces* enthalten branchen- und betriebstypenspezifische Dienste (z. B. spezifische Dienste für Fertigung, Finanzen und Telekommunikation).¹³

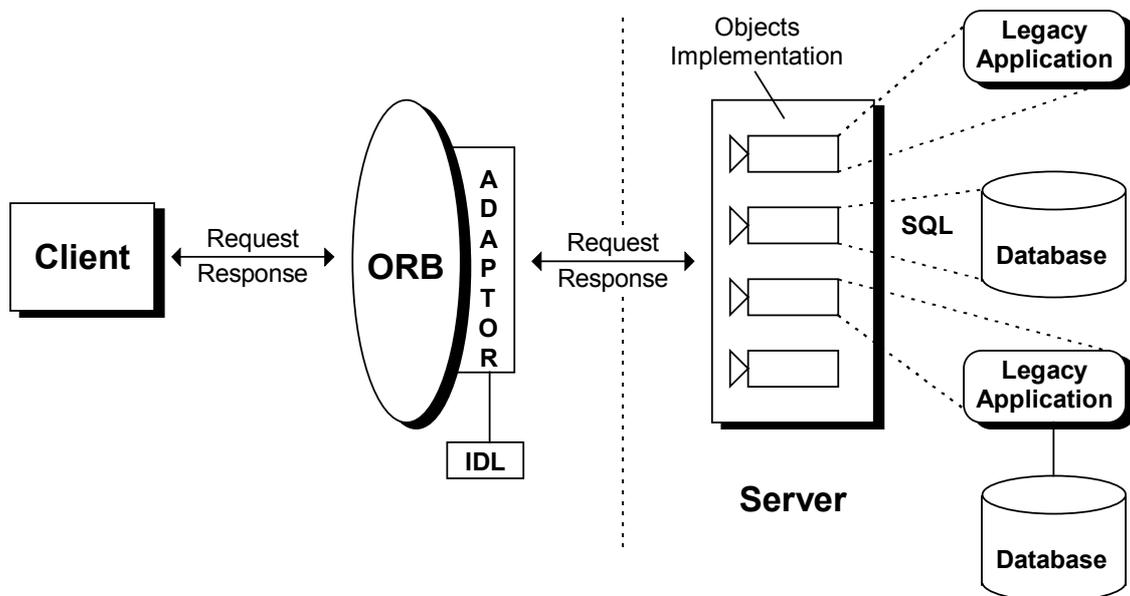


Abb. 1: Funktionsweise der Client/Server-Struktur in CORBA¹⁴

Abb. 2 stellt schematisch dar, welche Komponenten eine CORBA-basierte Architektur enthält und verdeutlicht die zentrale Rolle des ORB.

Die Applikationen und Daten in Form der Geschäftsobjekte werden nicht standardisiert, sondern individuell entwickelt, daher spricht man hier nicht von speziellen Diensten, sondern von *Application Interfaces*.

Abb. 2 deutet weiterhin an, daß Application Objects und Transaction Objects keine „reinen“ Objekte sein müssen, sondern daß auch nicht-objektorientierte Anwendungen als Objekte gekapselt und über den ORB mittels IDL standardmäßig angesprochen werden können.

¹³ Vgl. z. B. Vinoski, Steve: CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments, Cambridge, Mass., 1997; to appear in: IEEE Communications Magazine 2/1997, URL: <http://www.cs.wustl.edu/~schmidt/vinoski.ps.gz> (Stand: 15.7.1997), S. 2.

¹⁴ Vgl. Yang, Zhonghua; Duddy, Keith: CORBA: A Platform for Distributed Object Computing (A State-of-the-Art Report on OMG/CORBA), in: ACM Operating Systems Review 2/1996, S. 12.

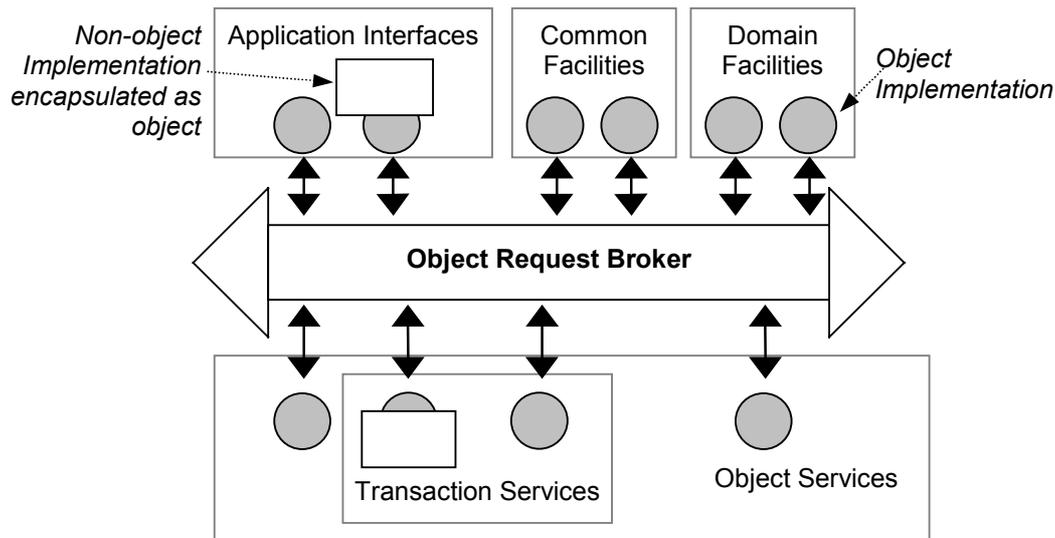


Abb. 2: Architektur für das Management verteilter Objekte¹⁵

3.3 CORBA-konforme Workflow-Architekturen

Prinzipiell sind drei Möglichkeiten vorstellbar, eine CORBA-konforme IT-Architektur bzw. deren Elemente zur Realisierung von Workflow-Systemen zu nutzen:

1. Der ORB kann als Middleware-Komponente die Kommunikation des Workflow-Systems mit externen Applikationen und Datenbeständen effizient unterstützen. Dies geschieht über die Nutzung des ORB als „Transportmittel“ und die Integration von Business Objects.
2. Über den ORB kann zusätzlich die Kommunikation der Client/Server-Komponenten des Workflow-Systems untereinander unterstützt werden. Darüber hinaus können Workflow-Systeme auf spezielle Dienste der CORBA-Architektur zurückgreifen. In diesem Fall sind Workflow-Systeme noch immer eigenständige Systeme, die quasi als spezielle Applikationen in den CORBA-Kontext eingebettet sind.
3. Workflow-Systeme werden als CORBA-Dienst(e) definiert und können wie die anderen Services und Facilities in CORBA genutzt werden. Workflow-Systeme sind dann nicht mehr eigenständige Systeme im Sinne einer monolithischen Software-Komponente, sondern gehen in einer CORBA Workflow Facility auf.

Abb. 3 positioniert diese Möglichkeiten in einem Koordinatendiagramm und illustriert auf diese Weise, daß mit zunehmender CORBA-Konformität auch eine steigende Effizienz der Workflow-Unterstützung in Aussicht steht.¹⁶ Zusätzlich ist zu bemerken,

¹⁵ In Anlehnung an Georgakopoulos, Dimitrios et al.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, a. a. O., S. 142.

¹⁶ Mit Effektivität kann hier nicht argumentiert werden, da CORBA die Workflow-Funktionalitäten, die den Anwender betreffen, nicht verändert. Die Wirkungsweise und das Einsatzgebiet der Workflow-Technik bleiben auch in CORBA-konformen Architekturen erhalten.

daß die einzelnen Möglichkeiten aufeinander aufbauen, nicht immer exakt voneinander zu trennen und durch fließende Übergänge gekennzeichnet sind (wie die Abbildung durch die Anordnung der Elemente andeutet).

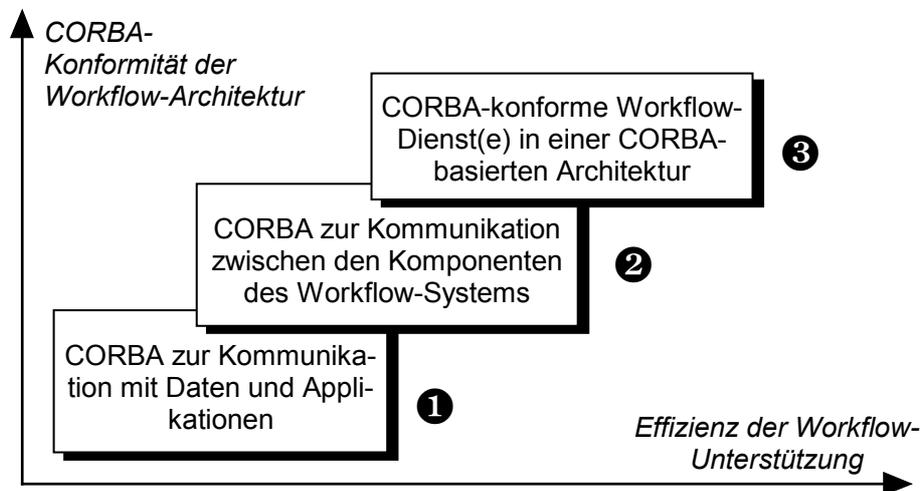


Abb. 3: Steigende Effizienz bei zunehmender CORBA-Konformität der Workflow-Architektur

1 CORBA zur Kommunikation mit Daten und Applikationen

Der entscheidende Vorteil der ersten Möglichkeit liegt in der erheblichen Reduktion des Aufwandes bei der Integration von Applikationen und Daten durch das Workflow-System: Die Kommunikation des Workflow-Systems mit Applikationen und mit Datenbanken kann durch den ORB sehr einfach abgewickelt werden, indem standardisierte, einheitlich in IDL spezifizierte Schnittstellen genutzt werden. Lediglich ein Remote Method Call, der die relevanten Parameter an das referenzierte Objekt übergibt, ist notwendig, um die Methoden des Objektes aufzurufen oder dessen Daten anzusprechen.¹⁷ Dies setzt jedoch voraus, daß die Applikationen und Datenbanken als Objekte angesprochen werden können, wenn nicht reine Objektumgebungen vorliegen. Wenn Umgebungen vorliegen, in denen Business Objects vorherrschen, nähert sich die Softwarefertigung und -integration der Idealvorstellung des Plug-&-Play-Prinzips.¹⁸ Auch der umgekehrte Weg muß möglich sein; d. h., die Geschäftsobjekte sollten über den ORB die Funktionalitäten des Workflow-Systems ansprechen können (z. B. Zustandsabfragen).¹⁹

17 Vgl. Miller, John A. et al.: WebWork: METEOR's Web-based Workflow Management System, a. a. O., S. 19.

18 Vgl. Rösch, Martin; Scharf, Tanja: Workflowsysteme und CORBA - Widerspruch oder Synergie?, in: Datenbank Fokus 4/1996, S. 18.

19 Diese Anforderung formulieren Jablonski et al. Von den aktuellen Workflow-Systemen wird die Einbindung von Geschäftsobjekten „in beiden Richtungen“ laut Jablonski et al. noch nicht unterstützt; vgl. Jablonski, Stefan et al. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, a. a. O., S. 283.

② CORBA zur Kommunikation zwischen den Komponenten des Workflow-Systems

Im Rahmen der zweiten Alternative wird die Kommunikation zwischen den Komponenten (Modulen) des Workflow-Systems (z. B. Engine, Transaktionsdienst, Authentifizierungsdienst) über den ORB abgewickelt. Diese Variante beinhaltet mehrere Vorteile:

- Durch die Nutzung des ORB zur Kommunikation ist der Zugriff auf Workflow-Funktionalitäten, d. h. Dienste des Workflow-Systems, von außen einfacher möglich, so daß auch andere CORBA-Dienste und externe Applikationen die Dienste des Workflow-Systems in Anspruch nehmen können (z. B. Nutzung des aufbauorganisatorischen Modells als Berechtigungsdienst).
- Die Module des Workflow-Systems können weitgehend plattform- und programmiersprachenunabhängig sowie verteilt realisiert werden, da die Kommunikation über standardisierte IDL-Schnittstellen abgewickelt wird.

Innerhalb des Workflow-Systems ist die Kommunikation standardkonform, so daß neben dem ORB auch weitere vorhandene Dienste der CORBA-Architektur genutzt werden können. Dies stellt einen entscheidenden Vorteil dar. Dienste der CORBA-Architektur, die in Workflow-Systemen genutzt werden könnten, sind z. B. die folgenden:²⁰

- *Event Service*: Events sind Nachrichten zwischen Clients und Servern, bei denen im Unterschied zu Method Calls nicht beide Kommunikationspartner (Client und Server) aktiv sein bzw. aktiviert werden müssen. Auf diese Weise ist eine asynchrone Kommunikation realisierbar. Relevant ist dies z. B. bei Systemausfällen oder bei stark schwankender Ressourcenverfügbarkeit. Jablonski et al. nennen als Beispiele für Anwendungsmöglichkeiten Zustandsmeldungen und die Synchronisation von verteilten Work Queues.²¹
- *Externalization Service*: Dieser Dienst ermöglicht es, den Zustand eines Objektes im Verlauf der Bearbeitung festzuhalten und an verschiedene Server weiterzugeben, z. B. an verschiedene ORBs oder andere Objekte, die Zustandsmeldungen auswerten (z. B. Statusmeldungen, umfassende Auswertungen über laufende Fälle). Darüber hinaus können Objekte auf einfache Art und Weise angezeigt, kopiert und zwischengespeichert werden (z. B. zu Zwecken der Datensicherheit).
- *Transaction Service*: Dieser Dienst zeichnet die im Verlauf der Abwicklung eines Falles durchgeführten Transaktionen auf, so daß ein Forward und Backward Reco-

20 Vgl. zu den einzelnen Diensten Miller, John A. et al.: WebWork: METEOR's Web-based Workflow Management System, a. a. O., S. 19 f. Auch Schulze nennt Beispiele für Dienste, auf die eine Workflow-Implementierung zurückgreifen kann; vgl. Schulze, Wolfgang; Böhm, Markus; Meyer-Wegener, Klaus: Services of Workflow Objects and Workflow Meta-Objects in OMG-compliant Environments, Position Paper for OOPSLA'96 Workshop on Business Object Design and Implementation, San José, CA 6.10.1996, S. 4. Weitere Beispiele finden sich bei ; vgl. Jablonski, Stefan et al. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, a. a. O., S. 283., S. 284 ff.

21 Vgl. Jablonski, Stefan et al. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, a. a. O., S. 284.

very sowie die Kompensation von Vorgangsschritten möglich werden. Auf diese Weise kann die Einhaltung der ACID-Anforderungen an Transaktionen sichergestellt werden.²²

Darüber hinaus ist über WWW-fähige ORB-Produkte²³ ein einfacher Anschluß an das Internet möglich, das somit als umfassendes Transportmedium verwendbar ist.

Bei diesem Architekturansatz ist zu bedenken, daß das Ausmaß, inwieweit die vorhandenen CORBA-Mechanismen genutzt werden, nicht festgelegt ist. Da das Workflow-System noch immer als eigenständiger Programmblock vorliegt, besteht die Gefahr, daß Funktionalitäten innerhalb dieses Blocks individuell programmiert werden, die auf der Basis von CORBA-Diensten realisiert werden könnten oder sogar schon realisiert sind (funktionale Redundanzen; z. B. systemspezifische Transaktionskonzepte).

Das Problem der Redundanz verschärft sich beim Einsatz mehrerer Workflow-Systeme sogar noch, da in diesem Fall gleiche Funktionalitäten in den unterschiedlichen Systemen mehrfach realisiert werden (z. B. Arbeitsvorratsverwaltung). Darüber hinaus ist Interoperabilität zwischen mehreren Workflow-Systemen lediglich dort vorhanden, wo auch die CORBA-Standards konsequent realisiert wurden.

Workflow-Systeme stellen hier noch immer eigenständige Programmsysteme dar, auch wenn sie wie ein CORBA-Dienst genutzt werden können. Will man den Idealzustand minimaler Redundanz und maximaler Interoperabilität erreichen, oder ihm zumindest sehr nahekommen, sowie gleichzeitig Konformität zum CORBA-Standard sicherstellen, so besteht im Grunde kein Platz für eigenständige Workflow-Systeme in der CORBA-Architektur. Eigenständige Workflow-Systeme stellen letztlich einen heterogenen Faktor in einer ansonsten homogenen, CORBA-konformen Architektur dar. Die Alternative 2 aus Abbildung 3 ist daher noch nicht gänzlich im Sinne der CORBA-Architektur und stellt praktisch eine Vorstufe auf dem Weg zu einer eigenständigen Workflow Management Facility (Variante 3 in Abbildung 3) dar.

③ CORBA-konforme Workflow-Dienst(e) in einer CORBA-konformen Architektur

Durch die Definition einer Workflow Management Facility im Rahmen einer CORBA-konformen IT-Architektur werden die Synergien zwischen den bestehenden CORBA-Komponenten und der CORBA-Grundstruktur noch intensiver ausgenutzt als in den bisher vorgestellten Varianten.²⁴ Über die Vorteile der vorhergehenden Stufen 1 und 2 hinaus ist hervorzuheben, daß hier funktionale Redundanzen weitestgehend entfallen und daß die Workflow-Komponente einen homogenen Bestandteil der gesamten IT-Architektur bildet.

22 Vgl. Schulze, Wolfgang et al.: Services of Workflow Objects and Workflow Meta-Objects in OMG-compliant Environments, a. a. O., S. 4.

23 Vgl. Jablonski, Stefan et al. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, a. a. O., S. 280.

24 Vgl. z. B. Jablonski, Stefan et al. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, a. a. O., S. 298 ff.

Ein auffälliger Unterschied zu den vorhergehenden Stufen dokumentiert sich darin, daß von einer Workflow Engine nicht mehr die Rede sein kann; die Funktionen der Workflow Engine werden direkt von bestimmten CORBA-Diensten übernommen.²⁵ Workflow-Systeme als monolithische Programmblöcke entfallen. Stattdessen wird eine Workflow Facility definiert, die lediglich diejenigen Dienste enthält, welche spezifisch für die Workflow-Technik sind und bisher noch nicht in CORBA vorhanden waren. Dienste, die schon in CORBA vorhanden sind (z. B. der Naming Service oder der Transaktionsdienst), müssen nicht nochmals in der Workflow-Komponente realisiert werden. Ein solcher Workflow-Dienst kann von allen Komponenten der IT-Architektur genutzt werden.

Der Vorschlag einer Workflow-Komponente, die als eine Facility innerhalb einer CORBA-basierten IT-Gesamtarchitektur implementiert wird, resultiert nicht nur aus den Vorteilen einer CORBA-konformen Workflow Facility, sondern auch aus einer „Lücke“ in der bisherigen CORBA-Architektur, die nicht alle für eine umfassende Workflow-Unterstützung notwendigen Komponenten enthält, obwohl sie in dieser Hinsicht definitiv ergänzungsfähig ist. Letztlich stellen aber auch die Nachteile aktueller Standardsysteme eine Motivation für den Vorschlag dar.

Bisher existieren lediglich Vorschläge²⁶, grobe Outlines innerhalb der Common Facilities Architecture Specifications²⁷ und ein Request for Proposal (RFP)²⁸ der OMG, wie eine solche Workflow Management Facility in die CORBA-Architektur integriert werden könnte. Nachfolgend soll grob umrissen werden, wie eine Workflow Facility innerhalb einer CORBA-konformen Architektur aussehen könnte.

Die Workflow Facility ist zuständig für das Management und die Koordination von Objekten, die Teil eines Workflows sind.²⁹ Dies beinhaltet letztlich alle Funktionen der Instanziierung und Steuerung, wie sie in herkömmlichen Workflow-Systemen die Workflow Engine übernimmt. Die Workflow Facility beinhaltet jedoch nicht mehr alle Funktionalitäten der Engine, sondern lediglich diejenigen, die noch nicht im CORBA-Kontext vorhanden sind. Bspw. muß sie Funktionalitäten für das Handling der Work Queues enthalten, während transaktionsbezogene Funktionen durch den Transaktionsdienst in CORBA abgedeckt werden könnten. Entscheidend bei dem Vorschlag einer Workflow Facility ist, daß Prozesse als Objekte definiert werden, so daß die Mechanismen der Ob-

25 Vgl. Schulze, Wolfgang: Towards an OMG-style Workflow Facility, Presentation to the CFTF, OMG Technical Meeting January 1997, Tampa, Florida, S. 5 ff.

26 Vgl. Schulze, Wolfgang et al.: Services of Workflow Objects and Workflow Meta-Objects in OMG-compliant Environments, a. a. O.

27 Vgl. Object Management Group (Hrsg.): Common Facilities Architecture, OMG Document formal/97-06-15, 1995-1997, Framingham, Mass., S. 5-1 ff.

28 Vgl. Object Management Group (Hrsg.): Workflow Management Facility RFP (Wolfgang Schulze, ed.), OMG Document cf/97-05-06, Mai 1997, Framingham, Mass.; Vorschläge waren bis zum 29.8.97 einzureichen und werden dann gemäß dem üblichen Procedere der OMG zu einem Request for Comment (RFC) verarbeitet. Das RFP enthält keine Spezifikationen, sondern umreißt lediglich grob und sehr allgemein die Anforderungen an eine Workflow Facility und nennt CORBA-Komponenten, deren Verhältnis zur Workflow Facility eindeutig zu klären ist.

29 Vgl. Object Management Group (Hrsg.): Common Facilities Architecture, a. a. O., S. 5-3.

jektorientierung greifen (z. B. Klassen, Vererbung) und sämtliche CORBA-Dienste auf diese Objekte anwendbar sind; so z. B. die folgenden:³⁰

- Die *Rule Management Facility* dient zur Definition und Verwaltung von Regeln über Workflow-Typen (z. B. Trigger zum Starten von Workflows und zur Fehlerkorrektur, Integritätsregeln, Event-Condition-Action-Muster).
- Die *Compound Document Facility* unterstützt Manipulationen an eingebetteten Objekten während der Workflow-Ausführung; dies ist insbesondere für dokumentenorientierte Workflows wichtig.
- Auf die Services der *Business Object Facility* wird zurückgegriffen, da Workflows spezielle Business Objects darstellen. Typische Services sind z. B. das Anlegen und Löschen von Objekten sowie das Einholen von Zustandsmeldungen.

3.4 Ein Praxisbeispiel: Die CORBA-basierte Architektur der BKM

Für die BKM war bei der Entscheidung für Entwicklung eines CORBA-basierten Workflow-Dienstes zum einen ausschlaggebend, daß in einer ansonsten CORBA-konformen IT-Architektur kein „monolithisches“, nicht-CORBA-konformes Workflow-System eingesetzt werden sollte. Es sollte eine grundsätzliche Homogenität zwischen IT- und Workflow-Architektur bestehen. Zum anderen bestehen zwischen dem Workflow-Konzept und dem CORBA-Standard zahlreiche Synergien, die sich produktiv nutzen lassen, wie nicht zuletzt auch andere Systeme zeigen (z. B. das an der University of Georgia entwickelte Workflow-System METEOR)³¹.

Abb. 4 zeigt die IT-Architektur der neuen Kernanwendung „BKMJoker“ der BKM. Die Workflow-Komponente ist in mehrere Dienste und Einzelmodule aufgeteilt worden, wobei der ORB eine zentrale Rolle spielt:

- Die *Workflow Client Application* ist ein aus mehreren Komponenten bestehendes Programm, welches zur Administration und Überwachung der Workflow-Komponente sowie zur Definition der Aufbau- und Ablauforganisation dient. Das Programm ist für diese Aufgaben mit einer graphischen Benutzeroberfläche ausgestattet.
- Der *BKMDesk* ist diejenige Komponente der Architektur, die es dem Mitarbeiter ermöglicht, Vorgänge zu bearbeiten. Hierzu stellt der BKMDesk eine graphische Benutzeroberfläche zur Verfügung, in der die von ihm zu bearbeitenden Vorgänge in

30 Vgl. Schulze, Wolfgang: Towards an OMG-style Workflow Facility, a. a. O., S. 11, Jablonski, Stefan et al. (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, a. a. O., S. 288, Object Management Group (Hrsg.): Workflow Management Facility RFP, a. a. O., S. 22 f. Zu den Task Management Common Facilities vgl. insbes. Object Management Group (Hrsg.): Common Facilities Architecture, a. a. O., S. 5-1 ff.

31 METEOR ist ein Forschungsprojekt, das maßgeblich vom Large Scale Distributed Systems (LSDIS) Laboratory der University of Georgia betrieben wird. Im März 1997 wurde das erste Release in der Connecticut Healthcare Research and Education Foundation eingesetzt, geplant ist (Stand: März 1997) auch der Einsatz bei Boeing; URL: <http://lsdis.cs.uga.edu/demos/workflowindex.html>.

Form von Work Queues (z. B. Posteingang und -ausgang) präsentiert werden. Bei der Auswahl eines Items aus der Work Queue wird die zugehörige Applikation aufgerufen und die relevanten Daten werden bereitgestellt.

- Die *System Client Application* ist im Gegensatz zum BKMDesk nicht für interaktive Arbeitsschritte zuständig, sondern für automatische Aktionen, die für den Bearbeiter nicht sichtbar sind (z. B. Verarbeitung durch ein Programm oder einen Drucker).

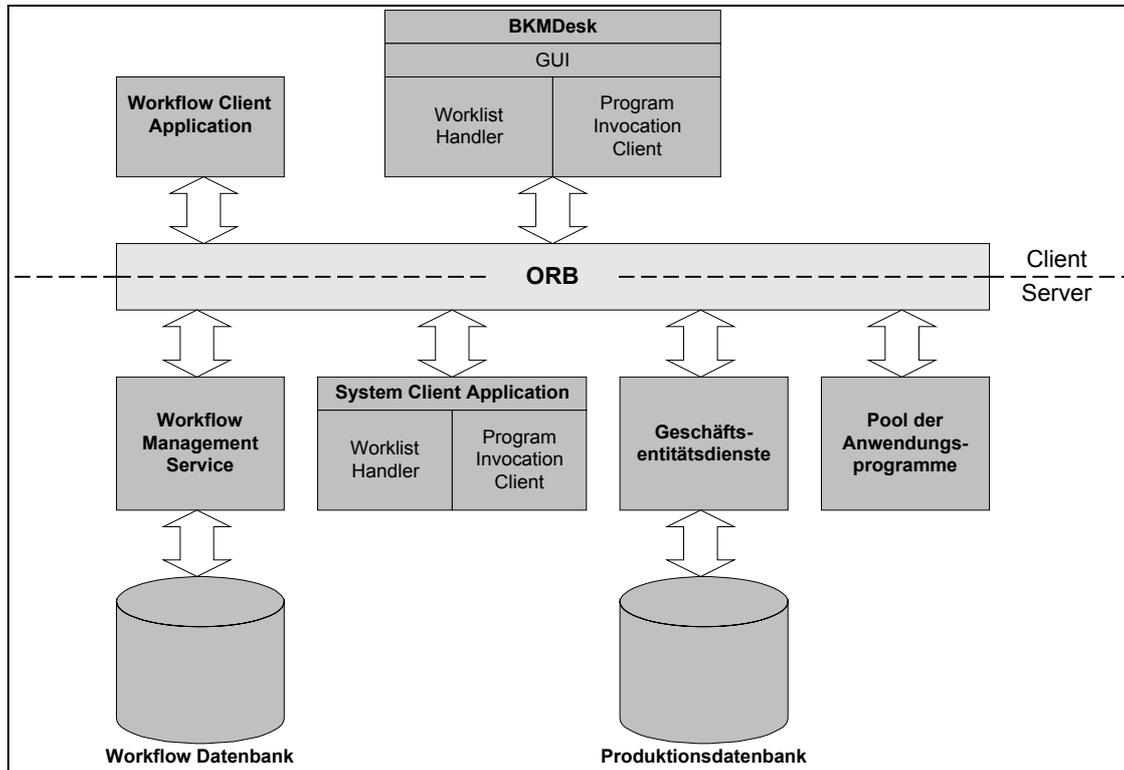


Abb. 4: Systemarchitektur der „Neuen Kernanwendung“ der BKM (vereinfacht)³²

- Der *Workflow Management Service* stellt dem Program Invocation Client des BKMDesk oder der System Client Application Informationen zur Verfügung, um die relevanten Applikationen und Daten aus dem Pool der Anwendungsprogramme zu selektieren und zur Ausführung zu bringen. Er fungiert als Dienst zur Instanziierung und Steuerung der Vorgangsabwicklung.
- In der *Workflow-Datenbank* werden die Instanzen und laufenden Falldaten abgelegt.
- Der *Pool der Anwendungsprogramme* enthält alle Anwendungsprogramme, die zur Abarbeitung der Workflows notwendig sind. Die Anwendungsprogramme werden zur Laufzeit entnommen und im System auf Clients und Servern verteilt.

³² Wolfrath, Michael: Fachkonzept Workflow Management und Kompetenzverwaltung Version 2.00 (BKM AG), Mainz 1997, S. 5.

- Auf die Fachdaten (z. B. Kontendaten, Kundendaten, Bausparverträge) greift die Workflow-Komponente über die *Produktionsdatenbank* zu. Der Zugriff auf diese Datenbank erfolgt ausschließlich über die *Geschäftsentitätsdienste*; diese stellen den Anwendungsprogrammen alle fachlichen Basisdienste für den Zugriff auf die Produktionsdatenbank zur Verfügung. Ein solcher Dienst ist z. B. ein Kontodienst, der alle Methoden zur Abfrage bzw. Manipulation der Konten einer Bausparkasse bereit hält.

Insgesamt wurde darauf geachtet, daß die einzelnen Komponenten so modular sind, daß sie bei Bedarf ausgetauscht (z. B. extern eingekauft) oder auch als eigenständige Produkte verkauft werden können. Diese Anforderung ergibt sich im Grunde direkt aus der Philosophie der CORBA-Architektur, die auf eine hohe Modularität abstellt. Letztlich steht und fällt der Erfolg von CORBA-basierten IT-Architekturen (und damit auch der CORBA-basierten Workflow-Komponente) mit der Zahl und Güte der zukaufbaren Module (z. B. Softwareprodukte für den Transaktionsdienst, aber auch einzelne Business Objects wie z. B. für die Kontenführung).

Ein besonderes Merkmal der neuen Kernanwendung stellt übrigens die WWW-Anbindung dar: Neue Applikationen werden im wesentlichen in Java realisiert, ein großer Teil der Workflow-Komponente ebenfalls. Sowohl die Workflow-Komponente als auch die Applikationen können daher in herkömmlichen WWW-Browsern ablaufen.

Die Workflow-Komponente der BKM weist alle Züge der Variante 2 aus Abbildung 3 auf. Die Workflow Facility, wie sie im RFP der OMG geplant ist, wird noch intensiver bestimmte CORBA-Mechanismen nutzen (z. B. Object Transaction Service, Rule Management Service), zu denen jedoch zum gegenwärtigen Zeitpunkt noch keine voll einsatzfähigen Produkte zur Verfügung stehen. Die bei der BKM verfolgte Variante 2 stellt jedoch eine sinnvolle Ausgangsbasis für eine spätere (vollständige) Migration zur dritten Stufe dar. Hier bleibt unter anderem abzuwarten, wie das Ergebnis des gegenwärtigen RFP zur Workflow Facility aussehen wird.³³

Die Möglichkeit, direkt mit der zweiten Variante zu beginnen, bot sich für die BKM an, weil praktisch die gesamte Anwendungslandschaft und -architektur neu entwickelt wurden. Dagegen sollten Unternehmen, deren IT-Architektur nur teilweise CORBA-konform ist, zunächst die erste Variante realisieren, so daß die Option für eine schrittweise Migration bis zur dritten Variante bleibt. Das schrittweise Vorgehen empfiehlt sich nicht zuletzt auch aufgrund der Unsicherheit, die mit dem CORBA-Standard derzeit noch verbunden ist. So sind derzeit noch nicht alle notwendigen Dienste in Form von Softwareprodukten auf dem Markt erhältlich, die für eine ausgereifte Workflow Facility notwendig wären. Der CORBA-Ansatz kann daher nur soweit verfolgt werden, wie es das aktuelle Angebot an CORBA-Produkten zuläßt.

³³ Laut dem Zeitplan der Workflow-RFP: Submissions bis August 97, endgültige Freigabe der Spezifikationen erst im April 1998. Produkte sind jedoch schon vorher zu erwarten, wenn es sich herausstellen sollte, daß die Spezifikationen Aussicht auf Erfolg besitzen, denn schließlich existieren schon Prototypen CORBA-basierter Workflow-Systeme. Vgl. Object Management Group (Hrsg.): Workflow Management Facility RFP, a. a. O., S. 24.

Literaturverzeichnis

- Casanave, Cory: Business-Object Architectures and Standards, URL: <http://www.dataaccess.com/bodtf/BOPaper.htm> (Stand: 12.5.97), Miami, Florida 1997.
- Czap, Hans: Common Object Request Broker (CORBA) Architecture, in: Information Management 1/1996, S. 71.
- Erdl, Günter; Schönecker, Horst G.: Studie: Geschäftsprozeßmanagement - Vorgangsteuerungssysteme und integrierte Vorgangsbearbeitung; München: FBO - Fachverlag für Büro und Organisationstechnik GmbH 1992.
- Georgakopoulos, Dimitrios; Hornick, Mark; Sheth, Amit: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure, in: Distributed and Parallel Databases 3/1995, S. 119-153.
- Götzer, Klaus: Workflow - Unternehmenserfolg durch effizientere Arbeitsabläufe; München: Computerwoche-Verl. 1995.
- Grell, Rainer: Elektronische Bearbeitung schwach strukturierter Vorgänge, in: Office Management 6/1995, S. 34-38.
- Jablonski, Stefan; Böhm, Markus; Schulze, Wolfgang (Hrsg.): Workflow-Management: Entwicklung von Anwendungen und Systemen - Facetten einer neuen Technologie, Heidelberg: dpunkt-Verl. 1997.
- Koch, Olaf G.; Zielke, Frank: Workflow Management - Prozeßorientiertes Arbeiten mit der Unternehmens-DV; Haar bei München: Markt und Technik, Buch- und Software-Verl. 1996.
- Miller, John A.; Planiswami, Devanand; Sheth, Amit P.; Kochut, Krys J.; Singh, Harvinder: WebWork: METEOR's Web-based Workflow Management System, Technical Report #UGA-CS-TR-97-002, Department of Computer Science, University of Georgia, März 1997.
- Object Management Group (Hrsg.): Common Facilities Architecture, OMG Document formal/97-06-15, 1995-1997, Framingham, Mass.
- Object Management Group (Hrsg.): Workflow Management Facility RFP (Wolfgang Schulze, ed.), OMG Document cf/97-05-06, Mai 1997, Framingham, Mass.
- Rösch, Martin: OMG-Standards und ihre Bedeutung für die Praxis, in: OBJEKTSpektrum 1/1994, S. 19-23.
- Rösch, Martin; Scharf, Tanja: Workflowsysteme und CORBA - Widerspruch oder Synergie?, in: Datenbank Fokus 4/1996, S. 16-22.
- Schulze, Wolfgang: Towards an OMG-style Workflow Facility, Presentation to the CFTF, OMG Technical Meeting January 1997, Tampa, Florida.
- Schulze, Wolfgang; Böhm, Markus; Meyer-Wegener, Klaus: Services of Workflow Objects and Workflow Meta-Objects in OMG-compliant Environments, Position Paper for OOPSLA'96 Workshop on Business Object Design and Implementation, San José, CA 6.10.1996.
- Schwab, Klaus: Koordinationsmodelle und Softwarearchitekturen als Basis für die Auswahl und Spezialisierung von Workflow-Management-Systemen, in: Vossen, G.; Becker, J. (Hrsg.): Geschäftsprozeßmodellierung und Workflow-Management; Bonn; Albany: Internat. Thomson Publ. 1996, S. 295-318.
- Vinoski, Steve: CORBA: Integrating Diverse Applications Within Distributed Heterogeneous Environments, Cambridge, Mass., 1997; to appear in: IEEE Communications Magazine 2/1997, URL: <http://www.cs.wustl.edu/~schmidt/vinoski.ps.gz> (Stand: 15.7.1997).
- Wolfrath, Michael: Fachkonzept Workflow Management und Kompetenzverwaltung Version 2.00 (BKM AG), Mainz 1997.
- Yang, Zhongua; Duddy, Keith: CORBA: A Platform for Distributed Object Computing (A State-of-the-Art Report on OMG/CORBA), in: ACM Operating Systems Review 2/1996, S. 4-31.

Bisher erschienen

Stand: Dezember 2000 – Den aktuellen Stand der Reihe erfahren
Sie über unsere Web Site unter <http://wi.uni-giessen.de>

Nr. 1/1996	Grundlagen des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 2/1996	Wettbewerbs- und Organisationsrelevanz des Client/Server-Konzepts.....	Schwicker/Grimbs
Nr. 3/1996	Realisierungsaspekte des Client/Server-Konzepts	Schwicker/Grimbs
Nr. 4/1996	Der Geschäftsprozeß als formaler Prozeß - Definition, Eigenschaften, Arten	Schwicker/Fischer
Nr. 5/1996	Manuelle und elektronische Vorgangsteuerung.....	Schwicker/Rey
Nr. 6/1996	Das Internet im Unternehmen - Neue Chancen und Risiken	Schwicker/Ramp
Nr. 7/1996	HTML und Java im World Wide Web.....	Gröning/Schwicker
Nr. 8/1996	Electronic-Payment-Systeme im Internet.....	Schwicker/Franke
Nr. 9/1996	Von der Prozeßorientierung zum Workflow-Management - Teil 1: Grundgedanken, Kernelemente, Kritik	Maurer
Nr. 10/1996	Von der Prozeßorientierung zum Workflow- Management - Teil 2: Prozeßmanagement und Workflow	Maurer
Nr. 11/1996	Informationelle Unhygiene im Internet.....	Schwicker/Dietrich/Klein
Nr. 12/1996	Towards the theory of Virtual Organisations: A description of their formation and figure.....	Appel/Behr
Nr. 1/1997	Der Wandel von der DV-Abteilung zum IT-Profitcenter: Mehr als eine Umorganisation.....	Kargl
Nr. 2/1997	Der Online-Markt - Abgrenzung, Bestandteile, Kenngrößen	Schwicker/Pörtner
Nr. 3/1997	Netzwerkmanagement, OSI Framework und Internet SNMP	Klein/Schwicker
Nr. 4/1997	Künstliche Neuronale Netze - Einordnung, Klassifikation und Abgrenzung aus betriebswirtschaftlicher Sicht	Strecker/Schwicker
Nr. 5/1997	Sachzielintegration bei Prozeßgestaltungsmaßnahmen.....	Delnef
Nr. 6/1997	HTML, Java, ActiveX - Strukturen und Zusammenhänge.....	Schwicker/Dandl
Nr. 7/1997	Lotus Notes als Plattform für die Informationsversorgung von Beratungsunternehmen.....	Appel/Schwaab
Nr. 8/1997	Web Site Engineering - Modelltheoretische und methodische Erfahrungen aus der Praxis	Schwicker
Nr. 9/1997	Kritische Anmerkungen zur Prozeßorientierung	Maurer/Schwicker
Nr. 10/1997	Künstliche Neuronale Netze - Aufbau und Funktionsweise	Strecker
Nr. 11/1997	Workflow-Management-Systeme in virtuellen Unternehmen	Maurer/Schramke
Nr. 12/1997	CORBA-basierte Workflow-Architekturen - Die objektorientierte Kernanwendung der Bausparkasse Mainz AG	Maurer
Nr. 1/1998	Ökonomische Analyse Elektronischer Märkte.....	Steyer
Nr. 2/1998	Demokratiopolitische Potentiale des Internet in Deutschland	Muzic/Schwicker
Nr. 3/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 1)	Delnef
Nr. 4/1998	Geschäftsprozeß- und Funktionsorientierung - Ein Vergleich (Teil 2)	Delnef
Nr. 5/1998	Betriebswirtschaftlich-organisatorische Aspekte der Telearbeit	Polak
Nr. 6/1998	Das Controlling des Outsourcings von IV-Leistungen	Jäger-Goy
Nr. 7/1998	Eine kritische Beurteilung des Outsourcings von IV-Leistungen.....	Jäger-Goy
Nr. 8/1998	Online-Monitoring - Gewinnung und Verwertung von Online-Daten.....	Guba/Gebert
Nr. 9/1998	GUI - Graphical User Interface.....	Maul
Nr. 10/1998	Institutionenökonomische Grundlagen und Implikationen für Electronic Business.....	Schwicker
Nr. 11/1998	Zur Charakterisierung des Konstrukts "Web Site".....	Schwicker
Nr. 12/1998	Web Site Engineering - Ein Komponentenmodell.....	Schwicker
Nr. 1/1999	Requirements Engineering im Web Site Engineering – Einordnung und Grundlagen.....	Schwicker/Wild
Nr. 2/1999	Electronic Commerce auf lokalen Märkten	Schwicker/Lüders
Nr. 3/1999	Intranet-basiertes Workgroup Computing	Kunow/Schwicker
Nr. 4/1999	Web-Portale: Stand und Entwicklungstendenzen.....	Schumacher/Schwicker
Nr. 5/1999	Web Site Security.....	Schwicker/Häusler
Nr. 6/1999	Wissensmanagement - Grundlagen und IT-Instrumentarium.....	Gaßen
Nr. 7/1999	Web Site Controlling.....	Schwicker/Beiser
Nr. 8/1999	Web Site Promotion	Schwicker/Arnold
Nr. 9/1999	Dokumenten-Management-Systeme – Eine Einführung	Dandl
Nr. 10/1999	Sicherheit von eBusiness-Anwendungen – Eine Fallstudie	Harper/Schwicker
Nr. 11/1999	Innovative Führungsinstrumente für die Informationsverarbeitung	Jäger-Goy
Nr. 12/1999	Objektorientierte Prozeßmodellierung mit der UML und EPK	Dandl
Nr. 1/2000	Total Cost of Ownership (TCO) – Ein Überblick.....	Wild/Herges
Nr. 2/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 1: XML-Grundlagen.....	Franke/Sulzbach
Nr. 3/2000	Implikationen des Einsatzes der eXtensible Markup Language – Teil 2: Der Einsatz im Unternehmen	Franke/Sulzbach
Nr. 4/2000	Web-Site-spezifisches Requirements Engineering – Ein Formalisierungsansatz	Wild/Schwicker
Nr. 5/2000	Elektronische Marktplätze – Formen, Beteiligte, Zutrittsbarrieren	Schwicker/Pfeiffer
Nr. 6/2000	Web Site Monitoring – Teil 1: Einordnung, Handlungsebenen, Adressaten.....	Schwicker/Wendt
Nr. 7/2000	Web Site Monitoring – Teil 2: Datenquellen, Web-Logfile-Analyse, Logfile-Analyzer	Schwicker/Wendt
Nr. 8/2000	Controlling-Kennzahlen für Web Sites.....	Schwicker/Wendt
Nr. 9/2000	eUniversity – Web-Site-Generierung und Content Management für Hochschuleinrichtungen.....	Schwicker/Ostheimer/Franke

Bestellung (bitte kopieren, ausfüllen, zusenden/zufaxen)

Adressat: Professur für BWL und Wirtschaftsinformatik
 Fachbereich Wirtschaftswissenschaften
 Licher Straße 70
 D – 35394 Gießen
 Telefax: (0 641) 99-22619

Hiermit bestelle ich gegen Rechnung die angegebenen Arbeitspapiere zu einem Kostenbeitrag von DM 10,- pro Exemplar (MwSt. entfällt) zzgl. DM 5,- Versandkosten pro Sendung.

Nr.	An
1/1996	
2/1996	
3/1996	
4/1996	
5/1996	
6/1996	
7/1996	
8/1996	
9/1996	
10/1996	
11/1996	
12/1996	

Nr.	An
1/1997	
2/1997	
3/1997	
4/1997	
5/1997	
6/1997	
7/1997	
8/1997	
9/1997	
10/1997	
11/1997	
12/1997	

Nr.	Anz
1/1998	
2/1998	
3/1998	
4/1998	
5/1998	
6/1998	
7/1998	
8/1998	
9/1998	
10/1998	
11/1998	
12/1998	

Nr.	Anz
1/1999	
2/1999	
3/1999	
4/1999	
5/1999	
6/1999	
7/1999	
8/1999	
9/1999	
10/1999	
11/1999	
12/1999	

Nr.	Anz
1/2000	
2/2000	
3/2000	
4/2000	
5/2000	
6/2000	
7/2000	
8/2000	
9/2000	

Absender:

Organisation _____

Abteilung _____

Nachname, Vorname _____

Straße _____

Plz/Ort _____

Telefon _____ Telefax _____ eMail _____

Ort, Datum _____ Unterschrift _____