

Rocks, Stones and Sand - Zur Granularität von Komponenten in Workflowmanagementsystemen

Rocks, Stones and Sand – Component Granularity in Workflow Management Systems

Prof. Dr. Jörg Becker, Dipl.-Wirt.Inform. Michael zur Mühlen

Zusammenfassung

Workflowmanagementsysteme können im betrieblichen Einsatz durch die Koordination von Akteuren, Aktivitäten, Daten und Anwendungssystemen die Effizienz der Prozeßausführung steigern. Die Stärke der Koordination variiert mit unterschiedlicher Granularität der in einen Workflow eingebundenen Applikationen. Um diese Varianz zu analysieren werden zunächst die Koordinationsmechanismen von Workflowmanagementsystemen klassifiziert und im folgenden anhand von drei Szenarien näher untersucht: Die Kopplung gesamter Anwendungssysteme auf grober Granularitätsebene (Rocks), die Kopplung von Modulen auf mittlerer Granularitätsebene (Stones) sowie die Kopplung von elementaren Funktionsbausteinen als Einbindung auf feiner Granularitätsebene (Sand). Die Auswirkungen der Szenarien auf die Koordinationsfunktionen von Workflowmanagementsystemen werden analysiert und Perspektiven für die Entwicklung von Workflow-Anwendungen aufgezeigt.

Summary

Workflow Management Systems increase the efficiency of process enactment through the coordination of activities, application systems, human resources and data. They are treated as a substantial enabler for the optimization potential discovered during business reengineering efforts. The effects of workflow coordination vary with the granularity of the components embedded in a workflow application, especially those applications that are invoked during the execution of a workflow. In order to explain this variance we analyze three different sizes of invoked applications with regard to their impact on workflow management: Entire application systems (rocks), modules (stones) and elementary functions (sand). The requirements for workflow management systems resulting from the various granularities are compared using the co-ordination functions of workflow management systems as a framework. Finally, perspectives for the development of workflow applications are outlined.

Stichworte

Workflowmanagement, Datenintegration, Granularität von Applikationen, Koordination, Anwendungsentwicklung, ERP-Software, Componentware

Keywords

Workflow Management, Data Integrity, Application Granularity, Co-ordination, Application Development, ERP-Software, Componentware

Rocks, Stones and Sand - Zur Granularität von Komponenten in Workflowmanagementsystemen

1 Motivation

Workflowmanagementsysteme (WfMS) werden aus Sicht von Unternehmen seit einigen Jahren als *Enabler* der durch Reorganisationsprojekte identifizierten Optimierungspotentiale wahrgenommen. Dennoch liegt die Zahl der produktiven Systeme im Vergleich zur Zahl der installierten Systeme insgesamt bei unter 30% [1]. Dies ist zum einen auf die Komplexität von Workflow-Einführungsprojekten zurückzuführen, bei denen zahlreiche organisatorische, technische und rechtliche Aspekte zu berücksichtigen sind. Zum anderen ist die Integration eines WfMS in die vorhandene technische Infrastruktur eines Unternehmens eine zeit- und kostenintensive Aufgabe.

Wir skizzieren drei Strategien für den unterschiedlichen Einsatz von WfMS in der Praxis und diskutieren für jede dieser Alternativen technische Realisierungsmöglichkeiten. Nach der Definition der wesentlichen Begriffe im folgenden Abschnitt werden Koordinationsmechanismen identifiziert, mit denen WfMS zur Zielerreichung bei Business Reengineering Projekten beitragen. Diese Koordinationsmechanismen werden dann anhand von drei Szenarien auf ihre Abhängigkeit von der Granularität der in einen Workflow eingebundenen Applikationen untersucht. Die Ergebnisse der Untersuchung werden abschließend synoptisch gegenübergestellt.

2 Basic Workflow Geology

Eine Basis der Realisierung integrierter Informationssysteme bildet die *Datenintegration*. Damit wird die Zusammenführung aller Daten eines Unternehmens unter Vermeidung eventueller Redundanzen (i. S. unerwünschter Mehrfachspeicherungen) und unter Berücksichtigung der Konsistenz (i. S. inhaltlicher Abhängigkeiten) der unterschiedlichen Daten bezeichnet. Eine erfolgreiche Datenintegration ermöglicht die unternehmensweite Entstehung und Nutzung von Daten auf effiziente Art und Weise [2].

Integrierte Anwendungssysteme haben jedoch nicht nur die integrierte Datenhaltung und -pflege sowie die Unterstützung der Ausführung von Einzelfunktionen zu leisten, sondern müssen dynamisch die Informationsflüsse entlang der Geschäftsprozesse unterstützen. In zwei Ausprägungen kann dabei die *Funktionsintegration* Anwendung finden. Zum einen können inhaltlich zusammenhängende Funktionen *vereinigt* werden (z. B. entwurfsbegleitende Kalkulation), zum anderen wird die *Verbindung* zeitlich-logisch zusammenhängender, aber asynchroner, d. h. aufeinanderfolgender Funktionen in Prozessen betrachtet (vgl. z.B. [3]). Die Realisierung der Funktionsintegration wird seit einigen Jahren im Rahmen diverser *Prozessmanagement*-Konzepte wie Lean Management [4], Total Quality Management [5], Supply Chain Management [6] oder Business Process Reengineering [7] verstärkt diskutiert.

Der Großteil der heute vorherrschenden Anwendungssysteme erfüllt die Anforderungen der Prozeßorientierung nur ungenügend, da sie in Form funktionaler Programmhierarchien aufgebaut sind und sich damit an der Erfüllung von Einzelfunktionen orientieren. WfMS bieten die Möglichkeit, diese funktional strukturierten Applikationen zu prozeßorientierten Anwendungen zu integrieren, indem sie Benutzer entlang der auszuführenden Geschäftsprozesse aktiv durch das System führen und ihnen die für die Funktionsausführung notwendigen Daten und Applikationen zur Verfügung stellen.

Unter einem *Workflow* verstehen wir den Teil eines Arbeitsablaufs, der die zeitlich-sachlogische Funktionsabfolge sowie Informationen über die an der Funktionsausführung beteiligten Daten und Ressourcen umfaßt. Ziel der Spezifikation eines Workflows ist die informationstechnische Unterstützung seiner Ausführung, wobei die Übergänge zwischen den einzelnen Aktivitäten unter der Kontrolle eines WfMS stehen. Dabei wird vom Inhalt der einzelnen Aktivitäten sowie der zu verarbeiteten Daten abstrahiert, sofern diese nicht den Arbeitsablauf selbst determinieren [8].

Aufgabe des Managements ist unter anderem die Koordination und Kontrolle von Leistungen, die von den Subsystemen einer Unternehmung erbracht werden. *Workflowmanagement* ist demzufolge die Koordination und Kontrolle von Workflows. Ein WfMS ist ein Anwendungssystem, welches das Management von Workflows ermöglicht bzw. unterstützt. Einige WfMS umfassen zudem noch Werkzeuge zur Modellierung von Workflowmodellen, während andere über Schnittstellen auf bestehende Werkzeuge zur Geschäftsprozeßmodellierung zurückgreifen (z. B. Adonis, ARIS, Bonapart, Nautilus, SOM/Pro). Eine *Workflow-Applikation* ist die Kombination aus einem oder mehreren WfMS und eingebundenen Anwendungssystemen, die im Unternehmenskontext unter Zugrundelegung eines Workflowmodells zur Unterstützung eines Arbeitsablaufs eingesetzt wird [9].

Viele Workflowmanagement-Projekte werden als Folgeprojekte im Rahmen von Geschäftsprozeßmodellierungs- bzw. -reorganisationsprojekten gestartet. Dabei gehen die erstellten Modelle der Geschäftsprozesse als Ausgangsdokumente in die Erstellung der Workflowmodelle ein. Während jedoch bei der Geschäftsprozeßmodellierung die *Organisationsgestaltung* im Vordergrund steht, wird bei der Workflowmodellierung die *informationstechnische Unterstützung* von Abläufen betrachtet (vgl. Abb. 1).

[Abb. 1]

3 Co-ordination and Efficiency

WfMS unterstützen die Ausführung von Arbeitsabläufen durch die Bereitstellung automatisierter Koordinationsmechanismen. Durch die Verlagerung dieser Koordinationsmechanismen aus bestehenden Anwendungssystemen und die Verringerung der manuellen Koordination ermöglicht der Einsatz von WfMS die effizientere Kontrolle und Anpassung von Steuerungsmechanismen im Unternehmen. Die Koordinationsaufgaben eines WfMS sind in Tabelle 1 dargestellt [11].

<i>Aktivitäten- koordination</i>	Das WfMS automatisiert die Übergänge zwischen den einzelnen Prozeßaktivitäten. Das implizite Wissen über die Abfolge der Aktivitäten kann entweder komplett auf das WfMS übertragen werden (Production Workflow, vgl. z. B. [12]) oder es wird in Teilen zur Laufzeit durch die beteiligten Akteure bestimmt (Ad-hoc- oder Collaborative Workflow). Die Workflow-basierte Koordination von Aktivitäten reduziert nicht wertschöpfende Tätigkeiten wie z. B. die Suche nach Informationen und unterstützt Lerneffekte durch die Explizierung des zugrundeliegenden Prozeßmodells.
<i>Koordination von Akteuren</i>	Das WfMS unterstützt die Zuweisung von Akteuren zu einzelnen Aktivitäten basierend auf einem Qualifikationsprofil (Rollenauflösung). Die verwendeten Koordinationsinstrumente sind die Benachrichtigung der Akteure sowie die Synchronisationsmechanismen der Arbeitslisten. Dieser Koordinationsaspekt beschleunigt die Identifikation von qualifizierten Bearbeitern und eliminiert entsprechende Suchaktivitäten. Weiterhin kann eine auf historisierten Daten basierende Rollenauflösung (z. B. „Wer war für den Kunden zuletzt verantwortlich?“) Konzepte wie Prozeßverantwortliche oder „One face to the customer“ unterstützen.
<i>Daten- koordination</i>	Im Rahmen einer Workflow-Aktivität stellt das WfMS die für die Aufgabenerfüllung relevanten Daten zur Verfügung. Zusammen mit Imaging-Technologien und Dokumentenmanagement-Systemen ist die effiziente Bereitstellung von Daten eines der bedeutendsten Argumente für den Einsatz von WfMS [13].
<i>Koordination von Anwen- dungssystemen</i>	Während der Ausführung von Aktivitäten stellt das WfMS dem Bearbeiter die notwendigen Anwendungssysteme für die Erfüllung seiner Aufgabe bereit und überwacht deren Ausführung (z. B. eine Textverarbeitung oder ein Eingabeformular).
<i>Kontrolle und Regulierung</i>	WfMS begünstigen die automatisierte Ermittlung, Analyse und bedargsgerechte Präsentation von Laufzeitdaten über Workflow-Instanzen. Diese Daten sind die Hauptquelle für Frühwarnmechanismen und können zusätzlich als Basis eines kontinuierlichen Prozeßmanagements (feedback engineering) genutzt werden.

Tabelle 1. Koordinationsmechanismen von WfMS

Die Ausübung dieser Koordinationsfunktionen durch ein WfMS ermöglicht die effizientere Ausführung von Geschäftsprozessen, da manuelle Medientransporte weitestgehend entfallen, Entscheidungsprozesse soweit wie möglich automatisiert und Überwachungsfunktionen durch das Workflowmanagementsystem wahrgenommen werden. Zur Messung der Effizienz der Geschäftsprozeßausführung bieten sich fünf Effizienzkriterien an (vgl. [14], Tabelle 2). Die einzelnen Koordinationsmechanismen unterstützen dabei in unterschiedlichem Maße die Effizienzkriterien. So sorgt z. B. das zentralisierte Monitoring und Controlling für eine Steigerung der Markteffizienz, da die Auskunftsfähigkeit gegenüber den Kunden verbessert wird.

<i>Prozeß- effizienz</i>	Die Optimierung von prozessualen Kriterien wie Durchlaufzeit (zu minimieren) oder Termintreue (zu maximieren).
<i>Ressourcen- effizienz</i>	Der ökonomisch effiziente Einsatz der für die Prozeßausführung verfügbaren Ressourcen.
<i>Delegations- effizienz</i>	Der effiziente Einsatz der Kompetenz übergeordneter Einheiten (Überblickswissen) und untergeordneter Einheiten (Detailwissen).
<i>Markteffizienz</i>	Die effiziente Positionierung des Unternehmens in seinem Verhältnis zu Markpartnern (zuverlässige Vorhersage von Lieferzeiten, transparente Kommunikation mit Kunden und Lieferanten, optimierte Abrechnungs- und Distributionsprozesse).
<i>Motivations- effizienz</i>	Motivation der Mitarbeiter zu unternehmenszielkonformem Verhalten.

Tabelle 2. Effizienzkriterien der Geschäftsprozeßausführung.

4 Invoked Applications

Eine *Invoked Application* ist eine bestehende Software-Lösung, die in eine oder mehrere Aktivitäten eines Workflowmodells eingebunden wird, von denen sie zur Laufzeit aufgerufen wird, um dem Anwender die Bearbeitung seiner Aufgabe zu ermöglichen [15]. Dabei können zwischen dem WfMS und der Invoked Application Daten ausgetauscht werden, die für den weiteren Verlauf der Workflow-Instanz von Bedeutung sind, sog. *workflowrelevante Daten*. Die Nutzdaten, welche die einzelnen Anwendungsprogramme verarbeiten, sind für die meisten WfMS transparent im Sinne von unsichtbar. Diese *Applikationsdaten* werden daher meist nicht in der Datenbank des WfMS gespeichert. Eine Ausnahme bilden hier sog. vorgangsmappenorientierte Ansätze, bei denen explizit die elektronische Nachbildung eines Aktenordners durch ein Netz von Bearbeitungsstationen geroutet wird. Systeme, die diesen Ansatz verfolgen, sind z. B. CSE/WorkFlow oder IABG ProMInanD. Einige WfMS ermöglichen dem Workflow-Modellierer zusätzlich die Spezifikation von Hilfsanwendungen (*support tools*), die vom Anwender im Bedarfsfall aufgerufen werden können, die zur Bearbeitung der Aufgabe jedoch nicht zwingend notwendig sind (vgl. z. B. [16]).

Für die Einbindung von Applikationen in Workflows stehen verschiedene Mechanismen zur Verfügung. Das Referenzmodell der Workflow Management Coalition (WfMC) sieht ein explizites Interface zwischen dem Ausführungskern eines WfMS (dem Workflow Enactment Service) und den Invoked Applications vor (Interface 3, vgl. [17]). Dabei werden über dieses Interface workflowrelevante Daten sowie Kontrollinformationen über den Status der Invoked Applications ausgetauscht (started, stopped, suspended, terminated, ended etc.). Einige Autoren betonen jedoch, daß das Wissen über die Invoked Applications keine zentrale Information der Workflow-Engine darstellt, sondern dezentral in den ausführenden Workflow-Knoten vorgehalten werden sollte (vgl. z. B. [18]). Daraus ergeben sich zwei unterschiedliche Architekturansätze für WfMS:

Beim *Engine-Ansatz* werden die Workflowmodelle in einer oder mehreren Engines des Workflowmanagementsystems verwaltet. Diese Engines steuern und überwachen die Ausführung der einzelnen Workflow-Instanzen. Das Wissen über die in den Workflow eingebundenen Anwendungssysteme liegt bei diesem Ansatz zentral vor. Dies ist von Vorteil, wenn die in einem Workflow angesprochenen Bearbeitungsstationen auf die gleichen Ressourcen zugreifen können oder wenn die Konfiguration der Arbeitsstationen identisch ist (z. B. beim Einsatz von *Thin Clients*). Eine Engine-Lösung verlangt, daß die ausführenden Knoten nach dem Bearbeitungsende die Kontrollhoheit wieder an die zentrale Workflow-Engine zurückgeben, die den Kontrollfluß auswertet und die nächste zur Ausführung anstehenden Aktivität an die entsprechenden Workflow-Knoten weiterleitet. Damit ist der Einsatz von mobilen Clients eingeschränkt, bei denen Bearbeiter Teile einer Workflow-Instanz autonom und dezentral bearbeiten und erst nach Ende der Bearbeitung wieder Kontakt mit dem zentralen System aufnehmen (z. B. im Außendienst mit Hilfe von Laptops).

Beim *verteilten Ansatz* kapseln die Workflow-Instanzen ihre Ablauflogik in Form von Workflow-Objekten. Diese Objekte wandern autonom durch ein Netz von Workflow-Knoten, wo sie dezentral ausgeführt werden. Informationen über die auszuführenden Applikationen kann in diesem Fall spezifisch in der Workflow-Instanz oder sogar in den ausführenden Knoten abgelegt werden. Im Gegensatz zum Engine-Ansatz ist im verteilten Ansatz die Unterstützung mobiler Clients einfacher möglich, da jedes Workflow-Objekt autonom den Fortschritt seiner Bearbeitung kapselt, ohne mit einem zentralen Server zu kommunizieren.

Die Einbindung von Invoked Applications in Workflowmodelle kann mit unterschiedlicher Granularität erfolgen. Wir untersuchen im folgenden eine Dreiteilung von eingebundenen Anwendungen im Hinblick auf das betriebswirtschaftliche Einsatzpotential und die Implikationen für das eingesetzte WfMS. Die größte Granularität (*Rocks*) bilden dabei Aktivitäten, die komplette Applikationen einbinden. Der Aufruf einzelner Module und Funktionsbausteine aus Anwendungssystemen bildet die mittlere Granularitätsebene (*Stones*), während eine Ausführung elementarer Funktionsbausteine selbst durch ein WfMS die feinsten Granularitätsebene darstellt (*Sand*). Die Kontrollhoheit über die Steuerungslogik des zugrundeliegenden Prozesses verlagert sich dabei mit zunehmender Applikationsgranularität von der Applikationssoftware auf das WfMS. Während bei grober Granularität das WfMS kaum Steuerungsfunktionen ausübt, wird im Falle sehr feiner Granularität die komplette Anwendungslogik in das WfMS verlagert.

5 Rocks

Eine grobe Applikationsgranularität liegt vor, wenn ein WfMS Prozesse auf Unternehmensebene unterstützt und dabei die eingebundenen Anwendungssysteme als ganze triggert, ohne in größerem Maße Nutzdaten mit diesen Systemen auszutauschen. Die Komplexität des gesamten Geschäftsprozesses tritt hier nicht zum Vorschein, da ein solcher Unternehmensworkflow über eine geringe Anzahl von Aktivitäten verfügt, die sich auf einem hohen Abstraktionsniveau befinden.

Mögliche Applikationen für eine Kopplung auf grober Ebene sind z. B. komplette Altanwendungen, Batch-Prozesse auf Großrechner-Ebene, Produktionsplanungs- und -steuerungssysteme (PPS-Systeme) sowie weitere WfMS, die spezifische Teile eines Unternehmensprozesses steuern. Dabei tritt für das steuernde WfMS ein Teil der typischen Workflow-Funktionalität wie z. B. die Rollenauflösung in den Hintergrund. Sie kann in diesem Zusammenhang vielmehr dazu genutzt werden, um Prozeßverantwortliche für einzelne Abschnitte eines unternehmensweiten Geschäftsprozesses zu bestimmen. Sinnvoll ist eine Kopplung mit grober Granularität vor allem für Unternehmensanwendungen, bei denen die Funktionalität aktueller WfMS für eine Steuerung von Elementarfunktionen nicht ausreicht. So weisen PPS-Systeme im Bereich der Ressourcenkoordination zwar große Ähnlichkeiten mit den Funktionen von WfMS auf, so daß sich die Frage stellt, ob die Funktionen eines PPS-Systems nicht durch eine flexibel konfigurierbares WfMS übernommen werden können, allerdings wird

beispielsweise die ex-ante Kapazitätsterminierung von PPS-Systemen von derzeitigen WfMS nicht oder nur unzureichend angeboten. Das PPS-System kann daher besser als „black box“ in ein Workflowmodell eingebunden werden, wobei lediglich Start/Stop-Informationen und zwingend notwendige Steuerungsinformationen übertragen werden.

Denkbar ist eine Kopplung mehrerer WfMS auf Applikationsebene, wenn ein „Master“-WfMS die Steuerung mehrerer bereichsspezifischer WfMS übernimmt. Diese Kopplung kann prinzipiell auch horizontal durch den Austausch von Steuerungsinformationen zwischen den einzelnen WfMS vorgenommen werden (z. B. über das Interface 4 der WfMC, vgl. [19, 20]). Sind jedoch Applikationen einzubinden, die sich einer Workflow-Automatisierung entziehen, so ist die Steuerung durch ein übergeordnetes WfMS eine denkbare Alternative.

Folgendes Beispiel erläutert die Kopplung von Applikationssystemen auf grober Granularitätsebene: Ein WfMS steuert den Auftragsbearbeitungsprozeß eines Serviceunternehmens. Auslöser eines Auftrags ist der Anruf eines Kunden in dem Call-Center des Unternehmens. Nach der Aufnahme der Kundendaten wird eine Referenz auf den Kundenauftrag durch das WfMS an die Disposition der Einsatzkräfte weitergeleitet. Hier wird eine Eigenentwicklung eingesetzt, welche die Tourenplanung vornimmt und das Scheduling für die Einsatzkräfte optimiert. Die Leistungserbringung selbst ist ein manueller Vorgang, während die sich daran anschließende Abrechnung mittels des SAP R/3 Moduls FI durchgeführt wird.

[Abb. 2]

Eine Technik für die Realisierung einer Workflowsteuerung auf Applikationsebene wird derzeit unter dem Namen Simple Workflow Access Protocol (SWAP) entwickelt [21]. SWAP hat die Steuerung von Workflows über das Internet zum Gegenstand. In Abb. 2 ist ein Workflow-Aufruf durch das SWAP-Protokoll schematisch dargestellt. Das im Draft-Status befindliche Protokoll, welches auf dem Hypertext Transfer Protocol (HTTP, vgl. [22]) und der Extensible Markup Language (XML, vgl. [23]) aufsetzt, sieht vor, daß ein *Observer* einen HTTP-Request an den Uniform Resource Identifier (URI) einer Process Definition schickt. Diese Anfrage (1) veranlaßt die URI zur Generierung einer Process Instance, deren URI an den Observer zurückgemeldet wird (2). Der Observer kann an diese URI einen „subscribe“-Request senden (3), damit die Process Instance den Observer über Ausnahmefälle benachrichtigt bzw. das Ende der Bearbeitung zurückmeldet. Eine Process Instance kann elementar sein, d. h. sie kapselt die Abarbeitung ihrer Aufgabe vollständig ohne Interaktion mit Ihrer Umgebung. Sie kann jedoch auch weitere Prozeßinstanzen (i. S. v. Subprozessen) aufrufen. In SWAP wird daher nicht zwischen Aktivitäten und Prozessen unterschieden. Eine Aktivität im Sinne unserer Definition ist in SWAP ein elementarer Prozeß. Im Beispiel ruft die generierte Process Instance ihrerseits einen Subprozeß auf, der z. B. durch ein anderes WfMS implementiert wird. Dazu generiert die Process Instance einen Activity Observer (4), der die gleiche Funktionalität wie ein Observer besitzt, zusätzlich jedoch der untergeordneten

Prozeßinstanz Informationen über die aufrufende Instanz liefern kann. Dies ist z. B. sinnvoll, wenn ein Akteur in einer untergeordneten Workflow-Instanz erfahren möchte, wer diese Workflow-Instanz aufgerufen hat. Im Beispiel sendet die übergeordnete Process Instance eine Nachricht an den Observer (5a), daß ein Subprozeß initiiert wurde, während der Activity Observer gleichzeitig an eine Process Definition URI den Request sendet, eine neue Instanz dieser Process Definition zu generieren (5b). Dies geschieht wie in der ursprünglichen Aufrufbeziehung (6), lediglich wird diesmal die URI der neuen Process Instance an den Activity Observer zurückgemeldet, dieser abonniert die Informationen über den Ausführungsstand der Process Instance (7) und wird von dieser über ihre Beendigung unterrichtet (8). Daraufhin bearbeitet die aufrufende Process Instance die restlichen (internen) Aktivitäten und meldet letztlich das Ende der Bearbeitung an den Observer (9).

Das folgende Beispiel illustriert diesen Vorgang (vgl. Abb. 3). Ein Kunde startet durch das Absenden eines Formulars auf einer WWW-Seite einen Auftragsbearbeitungsworkflow. Durch das Absenden des Formulars wird ein Observer generiert, der innerhalb des Unternehmens die eindeutige Steuerung dieses Kundenprozesses überwacht. Der Observer generiert eine Instanz des Auftragsbearbeitungsworkflows, erhält den URI der entsprechenden Instanz, generiert eine WWW-Seite, auf der der Kunde seinen Auftragsstatus verfolgen kann, und sendet eine E-Mail mit der Adresse dieser Seite an den Kunden. Sofern der Observer durch den Notifikationsprozeß über Statusänderungen im Auftragsbearbeitungsworkflow informiert wird, aktualisiert er die WWW-Seite für den Kunden. Nach einer Reihe von Schritten triggert der Auftragsbearbeitungsprozeß den Produktionsprozeß. Dieser wird durch ein PPS-System gesteuert, welches die Auftragsdaten als Steuerungsdaten erhält und ansonsten als „black box“ agiert. Die Informationen auf der für den Kunden sichtbaren WWW-Seite werden um die Meldung „Produktion ist gestartet“ aktualisiert. Nach erfolgreicher Beendigung des Produktionsauftrags meldet das PPS-System seine Beendigung an den aufrufenden Auftragsbearbeitungsprozeß (COMPLETE). Nach einer Reihe abschließender Schritte erhält der Kunde die Versandmeldung seines Auftrags ebenfalls auf der WWW-Seite angezeigt.

[Abb. 3]

Im SWAP-Protokoll benötigt der aufrufende Observer keine Kenntnisse über die Interna der aufgerufenen Prozesse bzw. Workflows. Dadurch eignet sich SWAP insbesondere für die Instanziierung von Workflows durch externe Prozeßteilnehmer über das Internet, z. B. im Rahmen von Electronic-Commerce-Anwendungen. Durch die rekursiven Aufrufbeziehungen sind sowohl Anwendungen mit einem groben Granularitätsgrad denkbar, als auch Kopplungen von einzelnen Aktivitäten mit sehr feiner Granularität. Da die zugrundeliegenden Protokolle HTTP bzw. XML jedoch einen gewissen Kommunikationsoverhead verursachen, dürfte SWAP aus Performancegründen vor allem für die beschriebene Kopplung von Applikationen mit grober Granularität anwendbar sein.

Die Datenkonsistenz bei der Kopplung von Systemen auf Applikationsebene ist, da vorwiegend Steuerungsinformationen ausgetauscht werden, nur wenig problematisch. Konsistenz und Zugriffsschutz für die Applikationsdaten liegen nicht im Aufgabenbereich des WfMS, sondern müssen durch die Invoked Applications gewährleistet werden. Die ausgetauschten Daten umfassen vor allem Steuerungsinformationen für die angesteuerten Informationssysteme (Start/Stop/Suspend etc.), Nutzdaten werden nur in geringem Maße übertragen. Werden größere Datenmengen benötigt, sollte ein Zeiger auf einen Speicherort (bzw. den URI der entsprechenden Daten) übergeben werden. Anwendungsentwickler, die Komponenten für einen Einsatz auf grober Granularitätsebene entwerfen, müssen daher für eine Workflow-Automation Schnittstellen zur Verfügung stellen, die einen Aufruf des Anwendungssystems durch das WfMS, die Abfrage des Systemstatus (Ready/Running/Terminated/Finished) und die Übergabe von Steuerdaten bzw. Zeigern auf Nutzdaten ermöglichen.

Die genannten Koordinationsfunktionen von WfMS sind bei einer Kopplung auf Applikationsebene von unterschiedlicher Bedeutung. Während die Koordination von Aktivitäten aufgrund der relativ geringen Anzahl von Workflow-Schritten nur eine mittlere Bedeutung besitzt, ist die Koordination der Anwendungssysteme von höherer Bedeutung. Die Aktenkoordination spielt aufgrund des hohen Abstraktionsgrades eine eher untergeordnete Rolle ebenso wie die Koordination der ausgetauschten Daten. Aufgrund der protokollierten Historiendaten können Controlling-Informationen wie die Durchlaufzeit von Aufträgen oder die Bearbeitungszeit in den einzelnen Unternehmensbereichen leicht ermittelt werden. Wenn die Historiendaten zudem an die in den operativen Systemen vorliegenden Nutzdaten gekoppelt werden (z. B. durch Verweise auf die im Rahmen einer Data-Warehouse-Anwendung vorliegenden Daten der operativen Systeme), ist im Rahmen eines *Drill-Down-Prozesses* die Analyse von Kundenaufträgen bis auf Ebene der elementaren Auftragsdaten möglich. Zur Identifikation von Prozeßschwachstellen im Sinne eines Feedback-Engineerings eignen sich die Daten jedoch weniger, da keine detaillierten Informationen über die durchgeführten Teilaktivitäten vorliegen.

6 Stones

Eine mittlere Granularität liegt vor, wenn ein WfMS Teile von Anwendungssystemen in einen Workflow-unterstützten Prozeß einbindet, z. B. Transaktionen oder Module einer ERP-Software oder aktivitätsspezifische Teile einer Standardapplikation (z.B. Terminierungskomponente einer Projektplanungssoftware). Dies unterstützt den Trend, sich von „full-featured“ Applikationen zu trennen und aufgabenspezifische Lösungen in Form wiederverwendbarer Komponenten im Unternehmen einzusetzen [24]. Die Workflow-Applikation ist hier eine Hülle um das WfMS selbst, die Granularität der Invoked Applications ergibt sich damit aus dem Umfang der einzelnen Funktionsblöcke des umgebenden Systems (Module, Funktionen, Prozeduren). Sofern das WfMS in die Komponenten einer Softwarefamilie eingebunden ist, handelt es sich um ein sogenanntes *Embedded WfMS* z. B. SAP Business Workflow, Oracle

Workflow, Peoplesoft Applications Workflow oder COSA/BaaN [25]. Kritischer Faktor beim Einsatz eines Embedded WfMS ist die Offenheit des Systems, d. h. die Fähigkeit, Anwendungsprogramme die sich außerhalb der Kontrollsphäre des umgebenden Softwaresystems befinden als Invoked Applications in die Workflowmodelle einzubinden. Ist diese Möglichkeit nicht gegeben, so sind Embedded WfMS nur sinnvoll in Umgebungen einzusetzen, die funktional vollständig von der ERP-Software abgedeckt werden.

Ein denkbare Einsatzgebiet für die Kopplung von Applikationsmodulen ist z. B. die Konstruktion eines PPS-Systems aus einzelnen Komponenten (vgl. z. B. [26]). Dabei werden die einzelnen Module des PPS-Systems (Stücklistenauflösung, Kapazitätsterminierung etc.) durch Standardsoftwarekomponenten implementiert (im genannten Fall durch Teile der Microsoft Office Produktfamilie), die durch ein zentrales WfMS gesteuert werden.

Eine Plattform für die Implementierung einer solchen Architektur liegt mit dem Common Object Request Broker Architecture-Framework (CORBA) der Object Management Group (OMG) vor, welches in die Object Management Architecture integriert ist. Kernstück der OMA ist der Object Request Broker (ORB), der gemäß der CORBA-Architektur aufgebaut ist. Er stellt einen Kommunikationsbus für Client- und Server-Objekte zur Verfügung. Die Schnittstellen für die Kommunikation werden mittels einer Interface Definition Language (IDL) spezifiziert. Durch Verwendung einer IDL-Hülle können somit Altanwendungen als CORBA-Objekte transparent in eine CORBA-basierte Systemumgebung integriert werden.

Zur Unterstützung des ORB existieren Systemdienste, sogenannte Object Services, welche die Kommunikation zwischen den verteilten Objekten unterstützen und die Handhabung von Objekten auf dem CORBA-Bus vereinfachen. So ermöglicht z. B. der Life Cycle Service das Erzeugen, Löschen, Kopieren und Verschieben von Objekten und erleichtert somit deren Handhabbarkeit. Die Common Facilites sind im Gegensatz zu den Object Services höherwertige Dienste, die insbesondere die Benutzerinteraktion durch die Spezifikation von grafischen Benutzerschnittstellen oder Druckausgaben unterstützen. Diese Dienste müssen somit nicht von jeder Anwendung neu spezifiziert werden. Domain Interfaces spezifizieren im Gegensatz zu den weitgehend anwendungsneutralen Object Services Dienste für bestimmte Anwendungsbereiche z. B. Telekommunikation, Gesundheitswesen und Handwerk.

[Abb. 4]

Für die Zwecke des Workflowmanagements wurde eine neue Common Facility spezifiziert, welche Workflow-Dienste für beliebige Anwendungen zur Verfügung stellt (vgl. Abb. 4). Damit wandelt sich ein entsprechend ausgestatteter ORB von einer technischen Plattform für die Verteilung bestehender WfMS zu einem integralen Baustein für ein verteiltes WfMS an sich. Die Dienste der Workflow Management Facility können dabei auf Dienste der Business Object Facility zurückgreifen. Business Objects repräsentieren dabei Anwendungsbausteine, die eine einfache Applikationsentwicklung durch die Kombination dieser Bausteine ermöglichen. Durch die Koordination der Business Objects mit Hilfe der Workflow

Management Facility ist die Entwicklung von Workflow Enabled Business Applications möglich, ohne dafür auf eine dezidiertes WfMS zurückgreifen zu müssen. Die derzeit vorliegenden Spezifikationen sind jedoch noch nicht detailliert genug, um eine tatsächliche Implementation zu unterstützen. Der jointFlow betitelte Vorschlag verzichtet auf die Spezifikation eines Workflow-Metamodells sondern legt statt dessen – angelehnt an das Interface 4 der Workflow Management Coalition – den Schwerpunkt auf die Kommunikation zwischen den Prozessen bzw. Aktivitäten eines verteilt auszuführenden Workflows (vgl. [29] und [30]). Er soll durch weitere Spezifikationen zur Definition von Prozessen bzw. zum Management von Ressourcen ergänzt werden.

Im Vergleich zur Kopplung von Applikationen auf grober Granularitätsebene ist die Datenkonsistenz bei einer Kopplung von Systemen mittlerer Granularität wichtiger. Auch hier können Zeiger auf Nutzdaten übergeben werden, da jedoch die Anzahl der Aktivitäten im Workflowmodell zunimmt, die autonome Steuerungslogik der einzelnen Applikationen jedoch abnimmt, entsteht für das steuernde WfMS mehr Aufwand für die Aktivitätenkoordination. Die tendenziell höhere Komplexität der Workflowmodelle führt zu einem Anstieg der Steuerungsdaten. Sofern die Invoked Applications nicht über dezidierte Schnittstellen zum Daten-Im- und -Export verfügen, sind durch das WfMS auch Datenkonversionen vorzunehmen. Die Datenintegrität verlagert sich damit zunehmend von der Ebene der Applikationen auf die Ebene des steuernden WfMS.

Aufgrund der beschriebenen Komplexitätssteigerung steigt im Fall der mittleren Applikationsgranularität die Bedeutung der Aktivitätenkoordination ebenso wie die Bedeutung der Koordination der eingebundenen Anwendungssysteme. Da die Zuordnung von Bearbeitern zu Aktivitäten sich hingegen nach dem Grad der Arbeitsteilung innerhalb der einzelnen Module richtet, ist diese Koordinationsaufgabe von mittlerer Bedeutung bzw. fallabhängig. Sofern die Abarbeitung einer Aktivität durch einen Bearbeiter erfolgt, kann eine klassische Rollenauflösung inkl. der Zuweisung eines Akteurs durch das WfMS erfolgen. Treten innerhalb einer Aktivität aufgrund der Beschaffenheit des aufgerufenen Anwendungssystems jedoch mehrere Bearbeiter in Aktion, so kann – vergleichbar mit der Kopplung auf Applikationsebene – die Rollenauflösung lediglich die Zuweisung eines Modulverantwortlichen zur Folge haben. Problematisch ist in diesem Fall die Pflege eines Berechtigungskonzeptes. Die Zuordnung von Bearbeitern zu Aktivitäten erfolgt im Rahmen der Rollenauflösung gemäß der spezifizierten Qualifikationsanforderungen. Allein die Zuordnung einer Aktivität zu einem Bearbeiter stellt jedoch noch nicht sicher, daß dieser auch alle notwendigen Berechtigungen auf Applikationsebene besitzt, um die Aktivität tatsächlich auszuführen, z. B. den Zugriff auf bestimmte Feldinhalte einer Datenbank. Hier ist erhöhte Aufmerksamkeit vom Workflow-Modellierer gefordert, da gegebenenfalls Berechtigungskonzepte redundant im WfMS und im Anwendungssystem gepflegt werden müssen.

Die Informationen, welche im Rahmen der Kontroll- und Regulierungsfunktionen des WfMS generiert werden, weisen im Fall der modulweisen Kopplung ein mittleres Abstraktionsniveau auf, welches sich für das Controlling der Prozesse wie auch für eine Analyse der Prozesse im Rahmen des Feedback-Engineering eignet. Für eine detaillierte Prozeßanalyse ist jedoch auch hier die Kopplung der Historiendaten mit den Daten der operativen Systeme notwendig.

Im Vergleich zur Kopplung auf grober Granularitätsebene müssen Komponentenentwickler im Szenario „Stones“ für eine größere Offenheit der bereitgestellten Systeme sorgen. Neben der Übergabe von Steuerungsdaten und Nutzdaten sind z. T. Eingriffe auf unterschiedlichem Abstraktionsniveau zu implementieren, z. B. das Starten der kompletten Anwendung, der Aufruf eines einzelnen Moduls und der Start einer einzelnen Transaktion.

7 Sand

Die feinste Granularität der aufgerufenen Applikationen liegt vor, wenn ein WfMS elementare Funktionsbausteine der aufgerufenen Anwendungsprogramme steuert. In diesem Fall liegt keine autonome Steuerungslogik der Invoked Applications mehr vor, das WfMS kann somit auch genutzt werden, um die vorgegebene Steuerungslogik einer Anwendungssoftware zu umgehen und ursprünglich funktional entwickelte Systeme an geänderte Geschäftsprozesse im Unternehmen anzupassen. Bedingung dafür ist die Zugänglichkeit der eingebundenen Anwendungssysteme auf elementarer Ebene, zum Beispiel durch Remote Function Calls.

Denkbar ist jedoch nicht nur, daß bestehende Systeme in eine Workflow-Anwendung integriert werden, sondern daß das WfMS als ein CASE-Tool eingesetzt wird, um dessen Kern durch die Spezifikation von aktivitätenspezifischen Komponenten eine Workflow-spezifische Applikation entwickelt wird. Ein Reihe von WfMS stellt dafür dem Anwender Funktionen zur Verfügung, mit denen kleinere Applikationen für die Einbindung in einen Workflow selbst gestaltet werden können. Dazu zählen beispielsweise Maskengeneratoren (vgl. Abb. 5) sowie einfache Skriptsprachen. Vielfach reicht die Funktionalität dieser Werkzeuge aus, um komplette Workflow-Applikationen zu entwickeln. Das WfMS übernimmt in solchen Fällen nicht nur die Koordinationsfunktionen zur Laufzeit der Workflow-Instanzen, sondern kann im Rahmen der Entwicklung einer Workflow-Anwendung als integriertes CASE-Tool für die Gestaltung von prozeßorientierten Unternehmensanwendungen gesehen werden. Anstelle von Invoked Applications stellen die innerhalb der Workflow-Aktivitäten ausgeführten Programme somit *Integrated Applications* dar.

[Abb. 5]

Ein Beispiel für die Realisierung einer Workflow-Applikation auf feinsten Granularitätsebene ist die Realisierung eines Workflows zur Beschaffung von einem neuen PC bei der internen EDV-Abteilung eines Unternehmens. Im Rahmen des Beschaffungsprozesses wird vom Antragsteller ein Formular am Bildschirm ausgefüllt und an die EDV-Abteilung gesandt. Dort wird der Rechner über eine WWW-Schnittstelle bei einem Zulieferer bestellt, und über Schnittstellen zu externen Datenbanken werden die Daten des neuen Systems in verschiedene

Datenbestände eingetragen (IP-Datenbank, Bestandsführung etc.). Sämtliche Datenoperationen werden über Masken realisiert, die mit dem Formulareditor des WfMS entworfen wurden.

Bei solchen integrierten Workflow-Anwendungen wird Kontrolle über die gesamte Datenhaltung vom WfMS übernommen. In diesem Fall werden nicht nur die workflowrelevanten Daten, sondern auch die Applikationsdaten über die Schnittstelle zwischen WfMS und Invoked bzw. Integrated Applications übertragen. Dementsprechend verlagert sich die Verantwortung für die Datenkonsistenz bzw. -integrität vollständig auf die Ebene des WfMS. Dieses sollte daher die zur Laufzeit verwalteten Datenbestände in einer zentralen Datenbank ablegen, um Integritätsproblemen vorzubeugen. Bezüglich des Berechtigungskonzeptes ist das Szenario „Sand“ weniger problematisch als das Szenario „Stones“, da das WfMS die Kontrolle über die Applikations- und Datenzugriffe der Benutzer behält. Somit kann der Zugriff auf Aktivitäten und Daten zentral über das Rollenkonzept des WfMS gesteuert werden, was die Administration vereinfacht.

Die Koordinationsfunktionen des WfMS haben bei der Koordination von Elementarfunktionen bzw. der feinsten Granularitätsebene einen anderen Stellenwert als bei einer Kopplung auf Modul- bzw. Applikationsebene. Die Aktivitätenkoordination gewinnt an Bedeutung, da die Komplexität der unterstützten Workflows durch die feine Granularität der einzelnen Aktivitäten wächst. Die Bedeutung der Koordination der eingebundenen Anwendungssysteme sinkt jedoch in dem Maße, in dem die externen Anwendungssysteme durch spezifisch auf die Workflow-Aktivitäten abgestimmten Kleinst-Applikationen ersetzt werden. Die Aktorenkoordination besitzt bei feinerer Applikationsgranularität größere Bedeutung, da in diesem Fall die Zuweisung von Bearbeitern zu Aktivitäten im Rahmen der Rollenauflösung stattfindet. Für die Zuweisung von Prozeßverantwortlichen zu einzelnen Abschnitten müssen nun proprietäre Funktionen des jeweiligen WfMS herangezogen werden. Im Bereich der Kontrolle und Regulierung steigt mit dem Anteil der elementaren Aktivitäten auch die Anzahl der protokollierten Laufzeitdaten. Dies ermöglicht einerseits ein sehr detailliertes Monitoring laufender Workflow-Instanzen, was z. B. in bezug auf die Auskunftsfähigkeit gegenüber Kunden und Lieferanten positiv ist. Für das Controlling auf Basis von Historiendaten ist der Abstraktionsgrad jedoch zu gering, so daß hier auf Hilfsmittel zurückgegriffen werden muß, z. B. Data-Mining-Tools oder spezielle Prozeßinformationssystemen (vgl. z. B. [31]).

8 Charting the Workflow Landscape

Die Granularität der in einen Workflow eingebundenen Applikationen hat unterschiedliche Auswirkungen auf die von einem WfMS bereitgestellten Koordinationsfunktionen. Mit feinerer Granularität der Applikationen nimmt die Intensität der Koordination durch das WfMS zu. Während auf Ebene der „Rocks“ die Systemkopplung im Sinne einer Applikationskoordination der primäre Fokus der Workflow-Automation ist, wird bei „Sand“ eine intensive Koordination auf allen beschriebenen Ebenen Aufgabe des WfMS. Je stärker Koordinationsmechanismen von bestehenden Applikationen bzw. manuellen Aufgabenträgern auf ein WfMS ver-

lagert werden können, um so höher sind die zu erwartenden Effizienzsteigerungen während der Ausführung eines Geschäftsprozesses. Die Größe der eingebundenen Applikationen determinieren somit nicht nur die Ausführungslogik des modellierten Workflows, sondern auch die von einem Workflow-Einsatz zu erwartenden wirtschaftlichen Resultate für das betroffene Unternehmen. Der mit feinerem Granularitätsgrad zunehmenden Nutzen durch die Koordinationsintensität des WfMS steht ein erhöhter Realisationsaufwand entgegen, der sich insbesondere in der steigenden Komplexität der Prozeßmodelle niederschlägt. Tabelle 3 gibt einen zusammenfassenden Überblick, welche der einzelnen Koordinationsfunktionen von WfMS bei wechselnden Granularitätsgraden von Bedeutung sind.

	Applikationsebene (Rocks)	Modulebene (Stones)	Funktionsebene (Sand)
Applikationskoordination	Kopplung von kompletten Applikationen	Kopplung von einzelnen Funktionen/Modulen	Kopplung von Elementarfunktionen
Autonomie der eingebundenen Anwendungen	Vollständig, nur die Abfolge der Anwendungen wird gesteuert	Mittel, Reihenfolge der Modulaufrufe wird gesteuert	Gering (Invoked Applications) / gar nicht (Integrated Applications)
Aktivitätenkoordination	Wenige Aktivitäten - geringe Komplexität der Workflowmodelle	Mittlere Anzahl von Aktivitäten - mittlere Komplexität der Workflowmodelle	Viele Aktivitäten - hohe Komplexität der Workflowmodelle
Aktorenkoordination	Zuweisung von Abschnittsverantwortlichen	Zuweisung von Abschnittsverantwortlichen und Bearbeitern	Zuweisung von Bearbeitern
Berechtigungskonzept	Im Anwendungssystem	Verteilt	Im WfMS
Datenkoordination	Austausch von Steuerungsdaten / Austausch von Verweisen auf Nutzdaten	Austausch von Steuerungsdaten / Austausch von Verweisen auf Nutzdaten	Austausch von Steuerungs- und Nutzdaten
Datenkonsistenz	Vom Anwendungssystem sicherzustellen	Systemübergreifend sicherzustellen	Vom WfMS sicherzustellen
Monitoring und Controlling	Controlling und Monitoring auf hohem Abstraktionsniveau	Controlling und Monitoring auf mittlerem Abstraktionsniveau	Detailliertes Monitoring und Controlling nur mit Tool-Unterstützung sinnvoll

Tabelle 3. Gegenüberstellung der Funktionen unterschiedlicher Koordinationsmechanismen bei unterschiedlicher Applikationsgranularität.

Aus den genannten Gründen kann es keine allgemeingültige Empfehlung für die „optimale“ Granularität von Workflow-Komponenten geben. Die Kopplung von „Rocks“ als auch von „Sand“ kann in entsprechenden Szenarien sinnvoll sein. Gerade bei mittlerem Granularität können durch die zwischen WfMS und Applikationssystem verteilten Berechtigungskonzepte Probleme entstehen, die entweder zu einer feineren Modellierung der Workflow-Prozesse oder zu einer gröberen Applikationsgranularität und damit zu einer Verlagerung des Workflow-Fokus hin zu einer systemkoppelnden Funktion führen. Auch eine Mischung der geschilderten Szenarien ist denkbar (vgl. Abb. 6). Bestehende größere Anwendungssysteme, sofern sie über keine weitergehenden Schnittstellen verfügen, können auf Ebene der „Rocks“ gekoppelt werden, die Einbindung von ERP-Software kann auf der Ebene von „Stones“ erfolgen, während für Neuentwicklungen oftmals die vorhandenen Fähigkeiten der WfMS in Form von Maskeneditoren und Skriptsprachen ausreichen.

[Abb. 6]

C Abbildungen

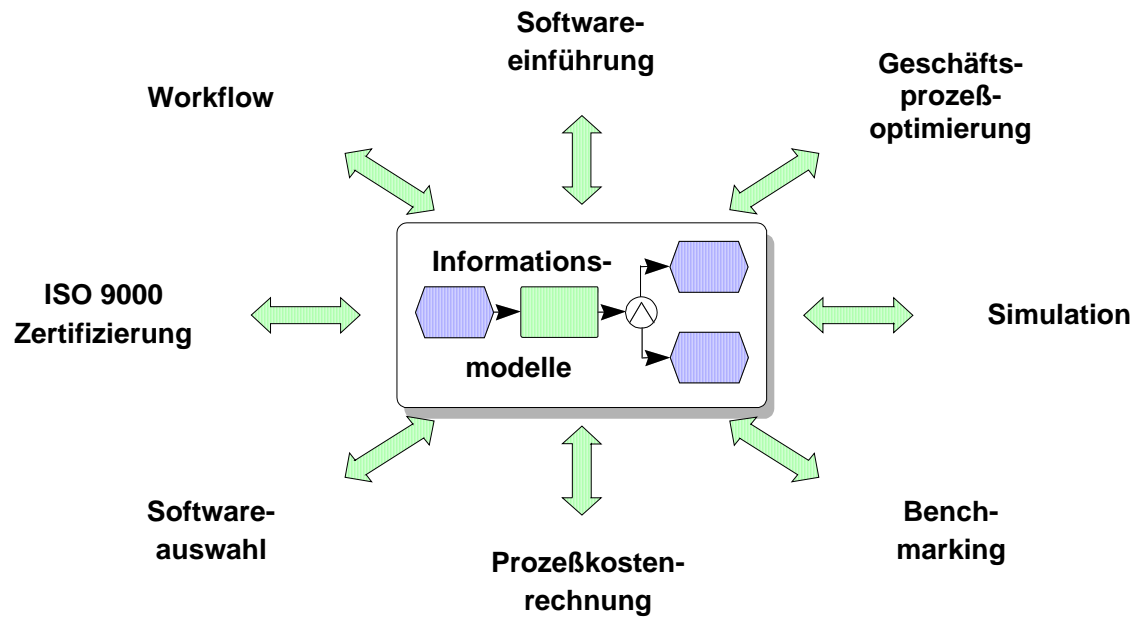


Abb. 1 Multiperspektivische Informationsmodellierung [10]

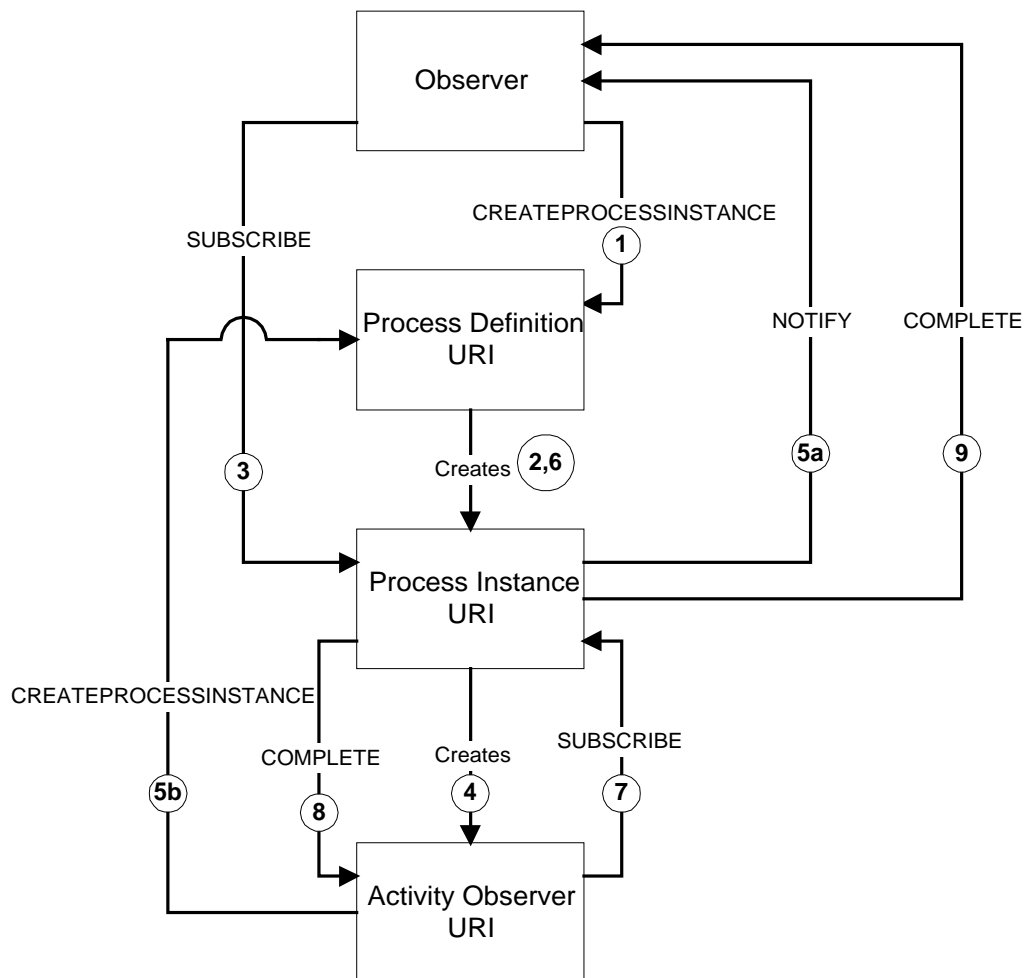


Abb. 2 Ablauf eines Workflow-Aufrufs durch das SWAP-Protokoll

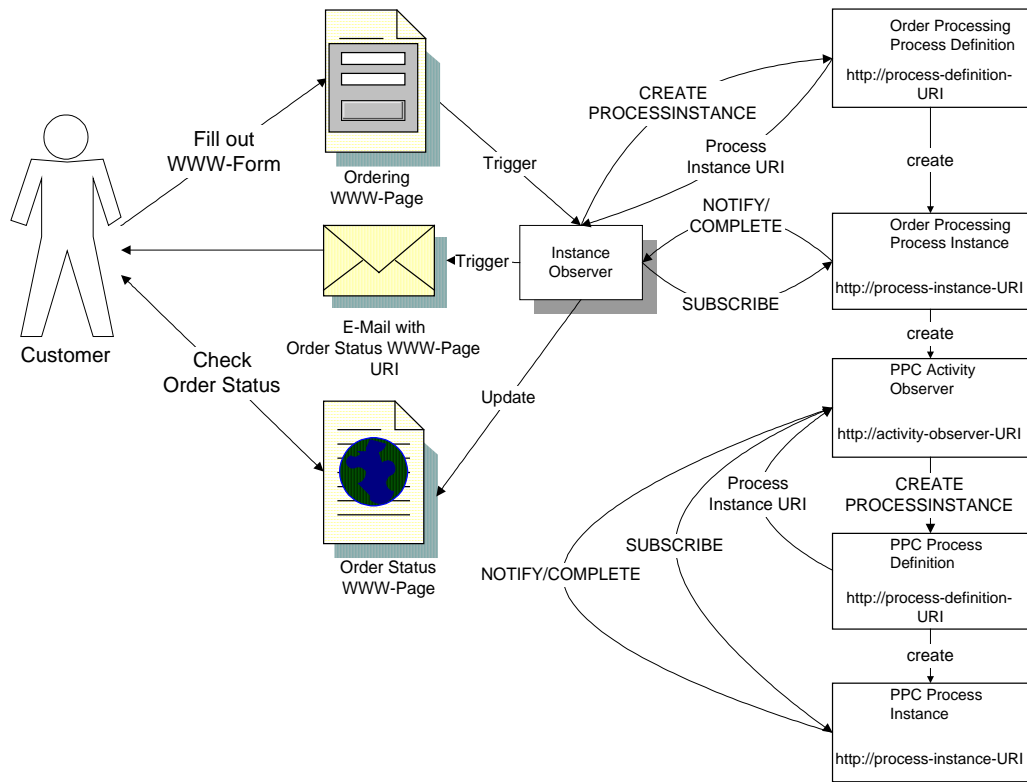


Abb. 3 Beispiel eines mit SWAP realisierten Unternehmensworkflows

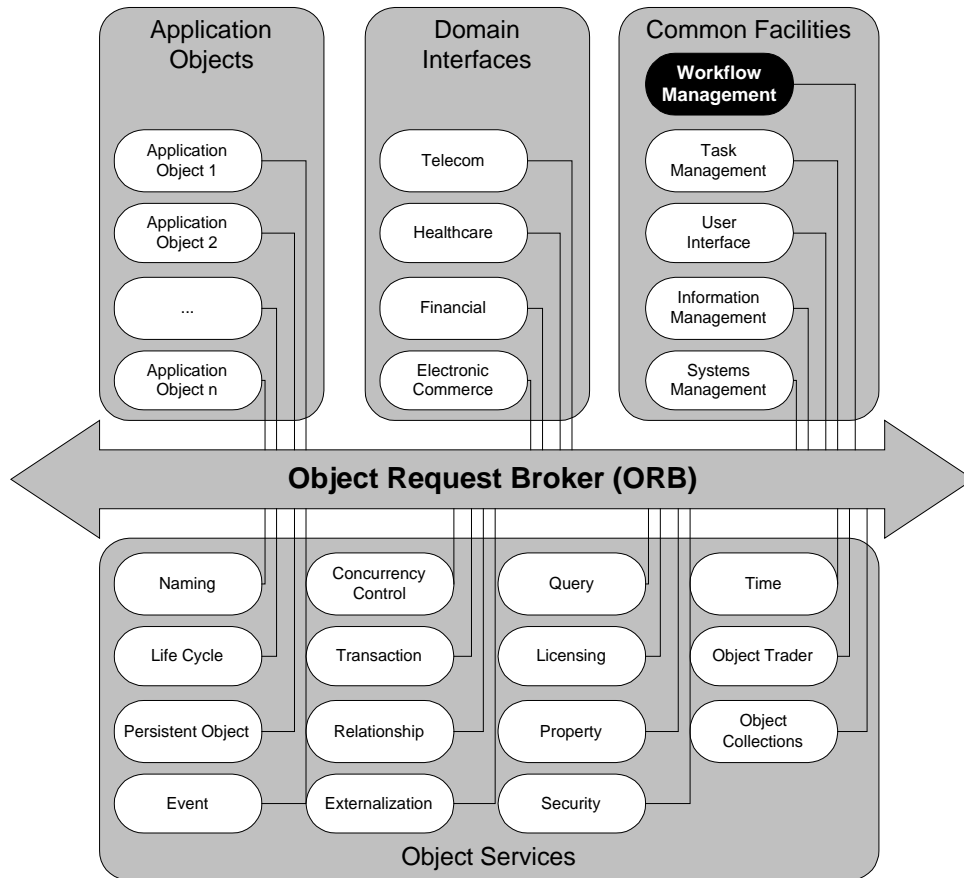



Abb. 4 Position der Workflow Management Facility im Rahmen der Object Management Architecture (in Anlehnung an: [27, 28])

Formular : A1 _ □ ×

Formular Ende Hilfe

Vorgang: V1

Case: 3-2 Antrag zur Mühlen 


Antrag auf Genehmigung einer Reisebeihilfe

Nachname:


Vorname:

Straße: Nr.:

PLZ: Ort:

Reisedatum vom: bis: 

Ziel der Dienstreise:

Kommentar: 

Antragsdatum:


Genehmigung:  Genehmigt am:

Abb. 5 Formular in Staffware 97

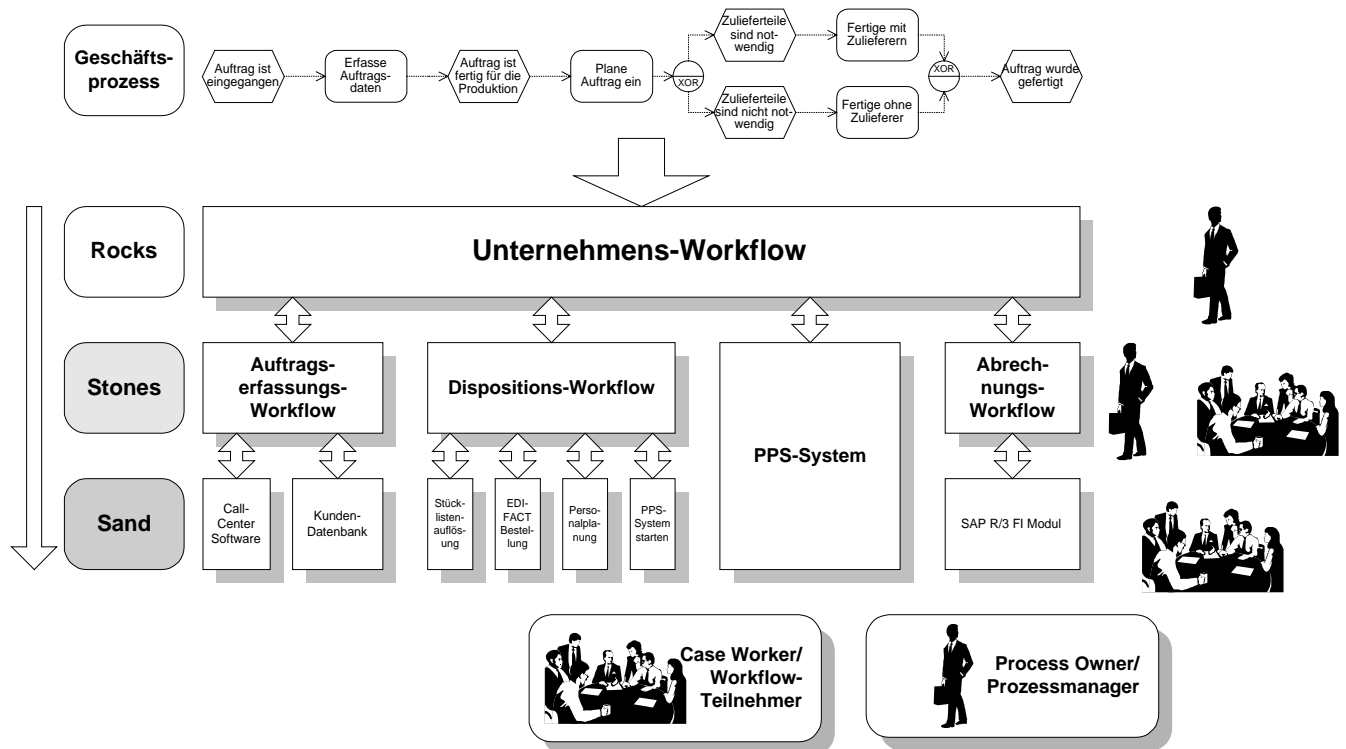


Abb. 6 Multi-level Workflow-Anwendung

D Literaturangaben

- [1] Delphi Consulting Group: Workflow Software Buyer's Perceptions Today. Volume WF96 – Number 1. Boston 1996.
- [2] Becker, J.: CIM-Integrationsmodell - Die EDV-gestützte Verbindung betrieblicher Bereiche. Berlin et al. 1991.
- [3] Becker, J.; Schütte, R.: Handelsinformationssysteme. Landsberg/Lech 1996.
- [4] Womack, J. P.; Jones, D. T.; Roos, Daniel: The Machine That Changed the World: The Story of Lean Production. Harper Perennial, New York 1991.
- [5] Ishikawa, K.: What is Total Quality Control? The Japanese Way. Prentice Hall, Englewood Cliffs 1985.
- [6] Hadjiconstantinou, E. (Hrsg.): Quick Response in the Supply Chain. Berlin et al. 1998.
- [7] Hammer, M.; Champy, J.: Reengineering the Corporation. A manifesto for business revolution. New York (NY) 1993.
- [8] zur Mühlen, M.: Evaluation of Workflow Management Systems Using Meta Models. In: Proceedings of the 32nd Hawai'i International Conference on System Sciences (HICSS '99). Wailea, Maui 1999.
- [9] Leymann, F. and Roller, D.: Workflow-based Applications. In: IBM Systems Journal, 36 (1997) 1, S. 102-123.
- [10] Rosemann, M.: Multiperspektivische Informationsmodellierung auf der Basis der Grundsätze ordnungsmäßiger Modellierung. Management & Computer, 4 (1996) 4, S. 229-236.
- [11] Becker, J.; v. Uthmann, Ch.; zur Mühlen, M.; Rosemann, M.: Identifying the Workflow Potential of Business Processes. In: Proceedings of the 32nd Hawai'i International Conference on System Sciences (HICSS '99). Wailea, Maui 1999.
- [12] Sheth, A.; Rusinkiewicz, M.: On Transactional Workflows. Data Engineering Bulletin 16 (1993) 2.
- [13] Fisher, L.: Excellence in Practice – Innovation and Excellence in Workflow and Imaging. Lighthouse Point 1997.
- [14] Theuvsen, L.: Business Reengineering. In: zfbf, 48 (1996) 1, S. 65-82.
- [15] Workflow Management Coalition: Terminology & Glossary. Document Number WFMC-TC-1011. Document Status Issue 3.0, April 1999, Winchester 1999.
- [16] IBM MQSeries Workflow: Getting Started with Buildtime Version 3.1.2. Document No. SH-12-6286-02. IBM Deutschland Entwicklungs GmbH, Böblingen 1998.

- [17] Workflow Management Coalition: Workflow Client Application Application Programming Interface (Interface 2 & 3) Specification. Document Number WFMC-TC-1009. Winchester 1997.
- [18] Wallnau, K.; Long, F.; Earl, A.: Toward a Distributed, Mediated Architecture for Workflow Management. In: Sheth, A. (Hrsg.): Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems. Athens, Georgia 1996.
- [19] Workflow Management Coalition: Workflow Standard – Interoperability Abstract Specification. Document Number WFMC-TC-1012. 20. October 1996, Version 1.0. Brüssel 1996.
- [20] Workflow Management Coalition: Workflow Standard - Interoperability Internet e-mail MIME Binding. Document Number WFMC-TC-1018. 18. September 1998, Version 1.1. Winchester 1998.
- [21] Swenson, K.: Simple Workflow Access Protocol (SWAP) Internet Draft (work in progress). <http://www.ics.uci.edu/~ietfswap/>
- [22] World Wide Web Consortium: HTTP – Hyper Text Transfer Protocol. <http://www.w3.org/Protocols/>
- [23] Walsh, N.: A Technical Introduction to XML. <http://www.xml.com/xml/pub/98/10/>
- [24] Engelhardt, A.: A Fresh Look at Integrated Business Applications – Neither Package nor Custom-Made! In: Proceedings of the 8th Australasian Conference on Information Systems, 1997, S. 517-528.
- [25] Becker, M.; Vogler, P.; Österle, H.: Workflow-Management in betriebswirtschaftlicher Standardsoftware. In: Wirtschaftsinformatik 40 (1998) 4, S. 318-328.
- [26] Möhle, S.; Braun, M.; Lauer, M.: Ein einfaches PPS-System mit Componentware. Arbeitspapier 2/1996 des Bereichs Wirtschaftsinformatik I der Universität Erlangen-Nürnberg. Nürnberg 1996.
- [27] Object Management Group: A Discussion of the Object Management Architecture. <http://www.omg.org/library/oma/oma-all.pdf>. Framingham 1998.
- [28] Orgfali, R.; Harkey, D.; Edwards, J.: The Essential Distributed Objects Survival Guide. New York et al. 1996.
- [29] jointFlow revised submission to the Workflow Management Facility RFP. OMG Document bom/98-06-07. Framingham 1998.
- [30] Errata relative to the OMG RFP #2 Submission Workflow Management Facility Revised Submission bom/98-06-07. OMG Document Number bom/98-07-15. Framingham 1998.
- [31] Rosemann, M.; Denecke, T.; Püttmann, M.: Konzeption und prototypische Realisierung eines Informationssystems für das Prozeßmonitoring und -controlling. Arbeitsbericht Nr. 49 des Instituts für Wirtschaftsinformatik. Münster 1996.

E Glossar

Ein *Prozeß* ist die inhaltlich abgeschlossene, zeitlich-sachlogische Folge der Funktionen, die für die Bearbeitung eines betriebswirtschaftliche relevanten Objektes notwendig sind. Dieses Objekt prägt den Prozeß, weitere Objekte können in den Prozeß einfließen.

Ein *Workflow* ist ein Prozeß, dessen Funktionsübergänge sich in der Kontrollsphäre eines Anwendungssystems, des Workflowmanagementsystems, befinden. Neben den Aktivitäten koordiniert das Workflowmanagementsystem dabei in unterschiedlicher Intensität die für die Workflowausführung notwendigen Daten, Applikationen, Akteure und Fristen.

Eine *Workflow-Applikation* ist die Kombination aus einem oder mehreren WfMS und den eingebundenen Anwendungssystemen, die im Unternehmenskontext unter Zugrundelegung eines Workflowmodells zur Unterstützung eines Arbeitsablaufs eingesetzt wird.

F Infoboxen

Object Management Group

Die Object Management Group (OMG) ist die führende Standardisierungsorganisation für objektorientierte Konzepte. Kernstück der OMG ist die Object Management Architecture (OMA), deren Ziel die Schaffung eines Rahmenkonzeptes für Anwendungen ist, welches die Wiederverwendbarkeit, Plattformunabhängigkeit und Interoperabilität von objektorientierten Software-Komponenten in verteilten und heterogenen Umgebungen ermöglicht. Weitere Informationen zur OMG finden sich unter <http://www.omg.org>.

Workflow Management Coalition

Die Workflow Management Coalition (WfMC) ist eine 1993 gegründete Non-Profit- Organisation, deren Hauptziel die Schaffung von Interoperabilitätsstandards zwischen Workflowmanagementsystemen ist. Die bekanntesten Publikationen der WfMC sind ein Glossar zum Thema Workflowmanagement sowie ein Referenzmodell, welches die Architektur von Workflowmanagementsystemen sowie die dazugehörigen Schnittstellen beschreibt. Der WfMC gehören mehr als 200 Mitglieder an, die sich aus Herstellern und Anwendern von Workflowmanagementsystemen, Beratungshäusern und Hochschulen zusammensetzen. Weitere Informationen zur WfMC finden sich unter <http://www.wfmc.org>.

Kernaussagen

Durch die Verlagerung von Koordinationsmechanismen aus bestehenden Anwendungssystemen ermöglichen von WfMS die effizientere Kontrolle und Anpassung von Steuerungsmechanismen.

Die Kontrollhoheit über die Steuerungslogik des zugrundeliegenden Prozesses verlagert sich mit zunehmender Applikationsgranularität von der Applikationssoftware auf das WfMS.

Wenn protokollierte Historiendaten an die in den operativen Systemen vorliegenden Nutzdaten gekoppelt werden (z. B. durch Verweise auf eine Data-Warehouse-Anwendung), ist im Rahmen eines Drill-Down-Prozesses die Analyse von Prozessen bis auf Ebene der elementaren Nutzdaten (z. B. Auftragsdaten) möglich.

Der mit feinerem Granularitätsgrad zunehmenden Nutzen durch die Koordinationsintensität des WfMS steht ein erhöhter Realisationsaufwand entgegen, der sich insbesondere in der steigenden Komplexität der Prozeßmodelle niederschlägt.

Es kann keine allgemeingültige Empfehlung für die optimale Granularität von Workflow-Komponenten geben.

G Autoreninformationen

Prof. Dr. Jörg Becker

Westfälische Wilhelms-Universität Münster

Institut für Wirtschaftsinformatik

Lehrstuhl für Wirtschaftsinformatik und Informationsmanagement

Steinfurter Str. 107

D-48149 Münster

Tel. +49-(0)251-8338100

Fax +49-(0)251-8338109

E-Mail: isjobe@wi.uni-muenster.de

<http://www-wi.uni-muenster.de/is>

Dipl.-Wirt.Inform Michael zur Mühlen

Westfälische Wilhelms-Universität Münster

Institut für Wirtschaftsinformatik

Lehrstuhl für Wirtschaftsinformatik und Informationsmanagement

Steinfurter Str. 107

D-48149 Münster

Tel. +49-(0)251-8338080

Fax +49-(0)251-8328080

E-Mail: ismizu@wi.uni-muenster.de

<http://www-wi.uni-muenster.de/is>