



**EAI 2004**

**Wilhelm Hasselbring, Manfred Reichert (Hrsg.)**

**EAI 2004 –  
Enterprise Application Integration**

**Tagungsband des GI-/GMDS-Workshops EAI'04,  
OFFIS, Oldenburg, 12. – 13. Februar 2004**

Proceedings



## **Vorwort**

Die heutige Flut von Informationsquellen im betrieblichen Umfeld bestimmt die immer weiter zunehmende Notwendigkeit der Integration von Anwendungssystemen und Datenbanken innerhalb der betrieblichen Datenverarbeitung und über Unternehmensgrenzen hinweg. Enterprise Application Integration (EAI) bezeichnet die Planung, die Methoden und die Software, um heterogene, autonome Anwendungssysteme unternehmensweit oder -übergreifend unter Einhaltung bestimmter Protokolle und unter Vermeidung von Abhängigkeiten zu integrieren.

Der Workshop EAI'2004 ist eine gemeinsame Veranstaltung des GI-Arbeitskreises „Enterprise Architecture“ und der GMDS-/GI-Arbeitsgruppe „KIS - Informationssysteme im Gesundheitswesen“. Auf dem Workshop wird die Thematik EAI aus übergreifender, ganzheitlicher Sicht betrachtet, und es werden die besonderen Herausforderungen aufgegriffen, diskutiert und der "State-of-the-Art" bestimmt. Das Workshop-Programm umfasst insgesamt 13 Beiträge aus Wissenschaft und Praxis, die sich mit Technologien sowie Methodik, Nutzen und Kosten des EAI und technischen Umsetzungsproblemen in der Praxis auseinandersetzen.

Neben Vorträgen zu EAI-Methoden und -Technologien gibt es Beiträge, die das Thema EAI aus Anwendungssicht näher beleuchten. Der Fokus liegt dabei auf Anwendungen aus dem Gesundheitswesen. Im Zentrum einer integrierten Gesundheitsversorgung sollte eine kooperative, sektorübergreifende und qualitätskontrollierte Behandlung der Patienten stehen. Diese wiederum ist nur effizient möglich, wenn die Informationssysteme der involvierten Leistungserbringer im Gesundheitswesen zusammenarbeiten. Berichte aus anderen Anwendungsbereichen, etwa der digitalen Fotografie und dem Elektrik-/Elektronik-Bereich, runden diese Betrachtungen ab, und ermöglichen einen Vergleich domänenspezifischer Anforderungen.

Zum Gelingen eines Workshops tragen viele bei. Wir möchten uns bei allen Vortragenden, insbesondere den eingeladenen Rednern, und ihren Koautoren bedanken. Ebenso herzlich danken wir den Mitgliedern des Programmkomitees sowie allen externen Gutachtern für ihre Sorgfalt und Mühe. Besonderer Dank gebührt auch dem OFFIS-Institut sowie Susanne Pedersen, Doris Reinders und Thorsten Teschke, die die Workshop-Leitung und das Programmkomitee mit großem organisatorischen Engagement unterstützt haben. Schließlich gilt unser Dank den Sponsoren des Workshops für ihre Unterstützung.

Wir begrüßen Sie, liebe Workshopteilnehmer, herzlich zur EAI'2004 in Oldenburg und hoffen, dass Sie das Workshop-Programm und die Abendveranstaltung abwechslungs- und erkenntnisreich finden werden.

Wilhelm Hasselbring

Manfred Reichert

**Veranstalter:**

GI-Arbeitskreis "Enterprise Architecture"  
<http://akea.iwi.unisg.ch/>

GMDS/GI-Arbeitsgruppe KIS - Informationssysteme im Gesundheitswesen  
<http://www.med.uni-marburg.de/stpg/ukm/lt/medinformatik/kis-ag.html>

**Workshop-Leitung:**

Wilhelm Hasselbring (Universität Oldenburg)

**Programmkomitee:***Leitung:*

Wilhelm Hasselbring (Universität Oldenburg)  
Manfred Reichert (Universität Ulm)

*Mitglieder des Programmkomitees:*

Stefan Conrad (Universität Düsseldorf)  
Peter Haas (FH Dortmund)  
Arne Koschel (IONA Technologies)  
Klaus Kuhn (Universität Marburg)  
Richard Lenz (Universität Marburg)  
Jörg Ritter (OSC - IM Systems AG)  
Joachim Schelp (Universität St. Gallen)  
Jürgen Schlegelmilch (OFFIS)  
Hans-Peter Steiert (DaimlerChrysler AG)  
Alfred Winter (Universität Leipzig)

**Lokale Workshop-Organisation:**

Susanne Pedersen (Universität Oldenburg)  
Doris Reinders (OFFIS)  
Thorsten Teschke (OFFIS)

## Sponsoren



Making Software Work Together



OLDENBURGER FORSCHUNGS- UND ENTWICKLUNGSMINISTERIUM  
FÜR INFORMATIK-WERKZEUGE UND -SYSTEME





## **Inhaltsverzeichnis**

### **Eingeladene Beiträge**

- Joachim Marz (CeWe Color AG): *Anforderungen der digitalen Fotografie an die Integration und Migration von Informationssystemen* 3
- Sean Baker (IONA Technologies): *Middleware Interoperability* 7

### **Wissenschaftliches Programm**

#### *Web Services*

- Ulrike Greiner, Erhard Rahm (Uni Leipzig): *Quality-Oriented Handling of Exceptions in Web-Service-Based Cooperative Processes* 11
- Thorsten Teschke, Holger Jaekel (OFFIS Oldenburg), S. Kriehoff, M. Langnickel (Kommunale Datenverwaltung Oldenburg), Wilhelm Hasselbring, Ralf Reussner (Uni Oldenburg): *Funktionsgetriebene Integration von Legacy-Systemen mit Web Services* 19

#### *Prozesse und Workflows*

- Uwe Küssner (neofonie Technologieentwicklung und Informationsmanagement GmbH): *Prozessorientierte B2B-P2P-Integration für Wissensmanagementanwendungen* 29
- Thomas Bauer (Daimler-Chrysler Forschungszentrum Ulm): *Integration heterogener Applikationstypen durch Workflow-Management-Technologie* 39
- Christian Pérez de Laborda, Christopher Popfinger, Stefan Conrad (Uni Düsseldorf): *DÍGAME: A Vision of an Active Multidatabase with Push-based Schema and Data Propagation* 49

### *Architektur und Semantik*

- Cornelia Boles, Jörg Friebe, Till Luhmann (OSC - IM Systems AG): *Klassifikation von Integrationsszenarien* 57
- Stephan Aier, Marten Schönherr (TU Berlin): *Enterprise Application Integration als Enabler flexibler Unternehmensarchitekturen* 69
- Manfred A. Jeusfeld (Tilburg University): *Integrating Product Catalogs via Multi-language Ontologies* 79

### *Anwendungsbereich Gesundheitswesen*

- Mario Beyer, Klaus Kuhn, Richard Lenz (Uni Marburg), Christian Meiler, Stefan Jablonski (Uni Erlangen): *IT-basierte Integration in medizinischen Versorgungsnetzen* 91
- Sascha Müller, Rainer Lay, Christian Meiler, Stefan Jablonski (Uni Erlangen): *Integration in der medizinischen Anwendung: Prozessbasierte Datenlogistik* 99
- Andreas Zimolong, Gregor Nöthen, Sylvia Thun, Kurt Becker (synaix Beratung für das Gesundheitswesen), Stefan Fritz (eliqos Business Consulting & Management GmbH): *EAI und ITIL im Krankenhaus - was kosten die IT-Leistungen?* 107

# **Eingeladene Beiträge**



# Anforderungen der digitalen Fotografie an die Integration und Migration von Informationssystemen

– Eingeladener Vortrag –

Joachim Marz

CeWe Color AG und Co OHG, Oldenburg  
{Joachim Marz}@cewecolor.de

## 1 Hintergrund – Der Fotofinishing-Markt

Der Standard-Geschäftsprozess in der Fotofinishing-Industrie, d.h. im Fotolaborgeschäft, war über Jahrzehnte recht stabil: Der Konsument gab seine Fotoarbeiten in der Fotoannahmestelle ab, das Labor holte die Aufträge ab, entwickelte die Arbeiten und brachte – in der Regel am nächsten Arbeitstag – die Aufträge zurück zur Fotoannahmestelle. In den *Ausprägungen* dieses Standardprozesses hingegen waren die Änderungen erheblich

- Vor 20 Jahren wurden Fotoannahmestellen in der Regel von Fotografen betrieben; der Marktanteil der Fachgeschäfte ist mittlerweile jedoch auf 20% gesunken; heute sind weit mehr als 50% der Fotoannahmestellen Drogerieketten und Warenhäuser.
- Während man vor 20 Jahren noch eine Mark für ein Bild im Format 9\*13 cm bezahlte, werden heute im Durchschnitt 5 Cent je Bild verlangt.
- Der Automatisierungsgrad ist erheblich gestiegen; heute erreichen optimierte Labore eine Produktion von etwa 70 Aufträgen je Lohnstunde.

Etwa zwei Drittel des Marktes in Europa teilen sich drei Anbieter (CeWe Color, gefolgt von Fuji und Kodak); Kodak ist jedoch vor kurzem – zumindest in Deutschland – aus dem Laborgeschäft ausgestiegen. CeWe Color betreibt 24 Labore in 7 europäischen Ländern. Im Jahr 2002 wurden etwa 150 Millionen Aufträge gefertigt. Durchschnittlich befinden sich 2,5 Artikelpositionen in einer Auftragstasche (Filmentwicklung, n Fotos, Fotoindex).

Seit wenigen Jahren wird die „klassische“ Analogie von der Digitalfotografie verdrängt, in der die Basis des Fotos nicht mehr das silberbasierte Negativ, sondern eine JPEG-Datei ist; der Rückgang im Verkauf von Filmen beträgt in Westeuropa gut 10% jährlich. Digitalfotografie bringt für den Konsumenten erhebliche praktische Vorteile. Die variablen Kosten für das „Schießen“ eines Bildes sind nahezu Null: Bilder können sofort auf dem Monitor der Kamera kontrolliert sowie anschließend am PC betrachtet, bearbeitet und archiviert werden. Die Preise für Digitalkameras sind bei vergleichbarer Optik nicht mehr viel höher als die Preise für „klassische“

Fotoapparate. (Zudem ist es heute offensichtlich „in“, digital zu fotografieren!) Der Markt der Digitalfotografie ist bereits in das Stadium des Massenmarktes eingetreten.

Erfreulicherweise hat sich aus Sicht der Fotolabore gezeigt, dass Konsumenten zumindest einen Teil ihrer Digitalfotos auf Papier haben möchten. Dieses ist von der Handhabung her den neuen Betrachtungsweisen überlegen. Fotos werden entweder zu Hause gedruckt (*home printing*) oder in einem Großlabor entwickelt. Per *home printing* lassen sich Fotos deutlich schneller erzeugen als per Großlabor (einige Minuten versus einen Labortag); per *home printing* erzeugte Fotos sind jedoch auch deutlich teurer als im Großlabor erzeugte Fotos. Die Relation ist in etwa analog zur Herstellart Minilab (Vor-Ort-Labor in der Fotoannahmestelle) versus Großlabor im Bereich der „klassischen“ Fotografie. Großlaborketten kennen zwei Methoden, die JPEG-Dateien in das Labor zu bekommen:

1. Über Offline-Endgeräte, in denen der Konsument den Speicherchip der Kamera auf eine CD „entlädt“ (Digifilm-Maker)
2. Per Internet, über die www-Adresse der Fotoannahmestelle, z.B. [www.schlecker.de](http://www.schlecker.de) oder auch – als „new kid on the block“ – [www.t-online.de](http://www.t-online.de). Die gefertigten Fotos werden dann entweder vom Konsumenten in der Fotoannahmestelle abgeholt oder mit per Post versandt.

Im Jahr 2002 basierten 15% des Umsatzes von CeWe Color auf digitalen Produkten; da der Umsatz mit dem Ausbelichten von JPEG-Dateien derzeit mit etwa 300% p.a. steigt, rechnet CeWe Color damit, in wenigen Jahren mehr als die Hälfte des Gesamtumsatzes mit digitalen Produkten zu machen.

## 2 Die IT-Systemlandschaft

Die Anforderungen an Informationssysteme im Fotofinishing-Markt sind sehr spezifisch; zudem ist der Gesamtmarkt nicht groß genug, um für Anbieter von Standardsoftware attraktiv zu sein. Von daher setzt auch CeWe Color Standardsoftware, d.h. SAP R/3, nur in den Bereichen Rechnungswesen (FI für Finanzbuchhaltung, CO für Controlling, EC-CS für Konsolidierung) sowie Einkauf/Materialwirtschaft (MM) ein.

Die Individualsoftware, die in wesentlichen Teilen in den letzten fünf Jahren neu entwickelt wurde und die auf Basis von Oracle läuft, dreht sich im Wesentlichen um die Auspreisung der Taschen in den (24) Laboren. Die hierzu benötigten Stammdaten (Kundenstamm, Artikelstamm, Preisstamm) werden auf einem Zentralrechner in Oldenburg vorgehalten und gepflegt; die (etwa 500) User sind über VPN-Verbindungen mit dem Standort Oldenburg verbunden.

In jedem Labor befinden sich Auspreisungsrechner, auf die die für den jeweiligen Produktionstag gültigen Stammdaten einmal täglich überstellt werden. Nachdem ein Labor einen Auftrag produziert hat, werden im Rahmen der Endsortierung der Taschen die zur Auspreisung benötigten Daten (Kundennummer, Zahl und Artikelnummer der in der Tasche befindlichen Artikel) per Barcode beziehungsweise elektronisch gelesen und über Algorithmen der jeweilige Taschenpreis errechnet; der Verkaufspreis, den der Konsument entrichtet, wird als Etikett auf der Tasche angebracht. Die in den Laboren entstehenden Bewegungsdaten (Produktionsstatus der

Tasche, [Händler-] Einkaufspreis, Verkaufspreis, Lieferscheine) werden über die VPN-Leitungen zur Weiterverarbeitung auf den Zentralrechner überstellt. Zentrale Applikationen sind insbesondere

- das Kundendienst-Auskunftssystem, mit dessen Hilfe die (etwa 100) Kundendienstmitarbeiter den Produktionsstatus eines Auftrags einsehen können
- das Faktura-Programm, mit dessen Hilfe zweimal pro Monat die Faktura für die Fotoannahmestellen erstellt wird
- Management-Statistiken für die Unternehmensleitung, die hoch aggregiert Marktentwicklungen aufzeigen
- Laborstatistiken zur Errechnung von Kennzahlen wie Produktivität, Papierausschuß, Lieferzeiten etc.
- Vertriebliche Statistiken (Vertriebsinformationssystem), über die die Verkaufszahlen in verschiedenen *Measures* (Verkaufspreis, Einkaufspreis, Taschenszahlen etc.) nach unterschiedlichsten Filter- und Sortierkriterien dargestellt werden können.

### **3 Anforderungen der digitalen Fotografie an die Integration und Migration von Informationssystemen**

Die Auswirkungen der digitalen Fotografie auf den Markt und seine Geschäftsprozesse spiegeln sich in den Anforderungen an die IT-Systeme wieder.

- Für die Bearbeitung der über das Web eingesandten JPEG-Dateien („Online Photo Service“) musste eine völlig neue Infrastruktur geschaffen werden, die aufgrund des sehr hohen Mengenwachstums skalierbar gestaltet werden musste. Für den Online Photo Service wurden ein zentraler Webserver inklusive der Applikation für das Auftragsmanagement sowie ein VPN, das die digital produzierenden Labore mit diesem Server verbindet, aufgebaut.
- Digitalfotografie erfordert neue Auspreisungsalgorithmen.
- Ein Großteil der „Digitalkunden“ ist auch Kunde im klassischen Bereich. Daher müssen die Auspreisungsdaten der Digitalfotoproduktion in die „klassische“ IT-Systemlandschaft integriert werden.
- Alle Statistiken, die teils über Jahr(zehnt)e entstanden sind, müssen auf den neu entstandenen Markt angepasst werden.
- Der sich intensivierende Kontakt mit Endkunden verlangt seitens CeWe Color offene Systeme (über Webbrowser zu bedienende Datenbanken); das Bedrohungspotential durch Hacker beziehungsweise Viren steigt unter anderem durch die Öffnung der zentralen IT-Systeme ständig.

- CeWe Color nutzt ein in die Jahre gekommenes Cobol-Programm für die Fakturierung. Dieses Programm steht zur Ablösung an. Die Entscheidung, ob ein Individualprogramm erstellt werden oder ob die Funktionalität im SAP R/3-System realisiert werden soll, wird derzeit vorbereitet. Die Anforderungen der Digitalfotografie haben erhebliche Anforderungen an die Fakturierung, da über die Mail Order – Variante des Online Photo Service in zunehmendem Maße Rechnungen direkt für *Endkunden* erzeugt werden müssen. Dies hat nicht nur strukturelle Auswirkungen; auch die Anforderungen an die *Performance* steigen erheblich.

Die letztgenannte Migrations- bzw. Integrationsaufgabe wird als Business Case im Vortrag beleuchtet.

Die Frage, ob eine neu zu gestaltende Funktionalität als Individualsoftware oder als Teil der Standardsoftware realisiert werden soll, beziehungsweise ob eine Mischung der beiden Formen sinnvoll ist, erscheint im vorliegenden Fall besonders komplex: Das Fakturaprogramm ist durch ein hohes Datenvolumen gekennzeichnet: Immerhin 150 Millionen Aufträge p.a. sind zu fakturieren, entsprechend etwa 400 Millionen Artikelpositionen. Zudem sind Darstellung und Aufbau der Faktura recht kundenspezifisch, also muss das Programm sehr flexibel sein; dies spricht eher für eine Individuallösung. Im CeWe Color-eigenen R3-Finanzbuchhaltungssystem hingegen werden die Daten nur in aggregierter Form benötigt. Zur Integration der Datenflüsse in besagtes Finanzbuchhaltungssystem erscheint eine Realisierung des Fakturaprogramms innerhalb des SAP R/3-Systems vorteilhafter. Von der SAP AG wird zudem eine Zusatzsoftware zur Massen-Fakturierung (kleinerer Beträge), eine *Billing Engine*, angeboten, deren Einsatzmöglichkeiten zu analysieren sind. Insbesondere die durch die Digitalfotografie verstärkt benötigten Fakturen für Endkunden sprechen für eine solche Lösung. Auf weitere Eckpunkte der Entscheidung wird im Vortrag detailliert eingegangen.

Der Autor hofft, durch die Präsentation einen interessanten Business Case beizutragen; nicht zuletzt erhofft er sich jedoch auch Hinweise auf Erfahrungen, die Workshop-Teilnehmer in vergleichbaren Entscheidungssituationen gemacht haben.

# Middleware Interoperability

- Invited Talk -

Sean Baker

IONA Technologies

**Abstract.** Middleware islands occur when an enterprise uses different middleware that doesn't communicate easily. They can arise because different middleware is used where it is most appropriate in an enterprise. Or the mix may have arisen for historical reasons, such as keeping up with new middleware developments, or because certain bought-in applications introduced their bundled middleware into the enterprise. All but the smallest enterprises face some serious challenges in integrating such islands. This talk will look at a number of issues, including:

- What middleware hides, but as importantly what it exposes.
- The options for middleware interoperability, and a comparison of these.
- How a Service Oriented Architecture (SOA) can help?
- Can the Web Services standards help? (For example, what benefit can be gained because WSDL separates the logical and physical contract?)
- The need for enterprise quality of service.

One of the interesting conclusions is the benefit to be gained by allowing a choice over which middleware is used at the transport level.



# **Wissenschaftliches Programm**



# Quality-Oriented Handling of Exceptions in Web-Service-Based Cooperative Processes

Ulrike Greiner, Erhard Rahm

Department of Computer Science, University of Leipzig  
{greiner, rahm}@informatik.uni-leipzig.de

**Abstract:** Web services are increasingly used to integrate heterogeneous and autonomous applications in cross-organizational cooperations. A key problem is to support a high execution quality of complex cooperative processes, e.g. in e-business or health care. One important aspect that has received little attention so far is the dynamic handling of exceptions during process execution. To address this problem, we propose a rule-based approach to automatically control and enforce quality constraints for web-service-based cooperative processes.

## 1 Introduction

The cooperation of companies and organizations leads to a rise of cooperative business processes integrating autonomous and heterogeneous applications of different organizations. Web services are increasingly used to facilitate such an application integration. Web services encapsulate applications and provide a machine-readable, XML-based interface for application calls [1], hence allowing the service providers to preserve their autonomy. Currently, web services typically realize only simple functions such as database queries. However, they can also be used to encapsulate complex legacy applications or entire workflows. To implement real-world cooperative processes a large number of web services of different complexity and from independent providers may have to be integrated. For instance, in a collaborative fulfillment process different partners cooperate to fulfill a customer's order (e.g. for assembling a PC) according to specified delivery time and other quality constraints. Other examples include treatment of a patient by different physicians and hospitals or a travel-booking process containing several services for booking a flight, hotel, etc. Ensuring that such processes reliably serve their purpose is challenging due to the high degree of autonomy and heterogeneity of the cooperation partners. It implies achieving a high quality of web service execution which is affected by various quality characteristics on services such as response time, cost, location, or constraints on service input and output parameters (e.g., price limits, product configurations or delivery deadlines).

Supporting quality of service for web services has found considerable interest recently. Several studies focus on the dynamic selection of the best provider for a particular web service (e.g. [2],[3]), others on dynamic resource management (e.g., load balancing) to support sufficiently fast web service execution (e.g. [2],[4]). [5] proposes the use of semantic service descriptions, e.g. to improve dynamic service selection. Several researchers studied cooperative business processes in the form of interorganization-

al workflows (e.g. [6],[7],[8],[9]) assuming a tight coupling between the cooperation partners. Newer studies address web-service-based cooperative processes (e.g. [10],[11],[12]) which offer more autonomy for the partners. However, most approaches lack flexible exception handling mechanisms to support the execution quality of cooperative processes. BPEL4WS (*Business Process Execution Language for Web Services*) is a proposed standard for defining business processes by integration of web services [12]. It supports a basic exception handling by checking whether predefined fault messages or time-outs occur for web services at particular process steps. Such exceptions are to be handled by calling compensating or alternative services. However, exceptions that are not covered by fault messages (e.g., violations of quality constraints) and which not only occur at predefined process steps cannot be handled adequately. Hence, a more flexible exception handling approach is needed to adequately support the execution quality and robustness of cooperative processes.

We present a new approach to dynamic exception handling in web-service-based processes that supports the specification of quality constraints for services, in addition to conditions a service may offer itself. A rule-based approach is used to handle exceptions such as the violation of constraints or other events (e.g. service faults) occurring during process execution. In section 2 we give an overview of the Web-Flow architecture implementing the new exception handling approach within a dedicated component that can be used together with different web service process engines. Section 3 presents a classification of the quality constraints to be supported. Dynamic exception handling is discussed in section 4; section 5 closes with an outlook on future work.

## 2 Web-Flow Architecture

The Web-Flow system aims at offering quality support for cooperative business processes integrating web services available locally or provided by external partners. It separates monitoring and exception handling functionality within a dedicated component, MaX (Monitoring and eXception handling). This is to allow a generic solution that can be used in combination with different process execution engines for definition and execution of cooperative processes (see Fig. 1). The process execution engine uses a web service environment to call local and external web services encapsulating simple application programs or entire workflows. Alternatively, external services may be called over the process engine of the partner. Services are assumed to provide a WSDL (Web Service Description Language, [13]) interface, if necessary through a wrapper (e.g. for workflows).

Cooperative processes may be specified according to two major cooperation models. In a *simple cooperation model*, a cooperative process corresponds to a workflow whose activities may refer to external web services. Thus, the providers of the integrated services do not know the overall process. In a *complex cooperation model*, the involved partners agree on a cooperative process (including the services used and the order in which they are called). The partners have an equivalent functionality and interact in a peer-to-peer fashion. This implies a reduced autonomy because changes on the cooperative process have to be coordinated between the partners. Web-Flow is to support both cooperation models: the first model can be supported by using the web service

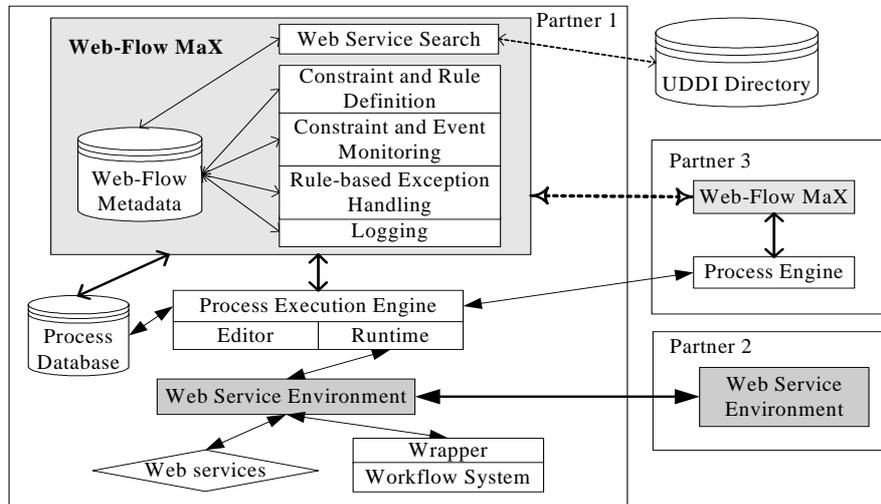


Fig. 1. Web-Flow architecture

functionality, the second by exploiting the direct communication between process engines (Fig. 1). Initially we focus on the simple cooperation model with one partner controlling the cooperative process. This approach supports maximal autonomy for service providers but implies that quality monitoring and exception handling have largely to be performed by the node controlling the cooperative process.

The Web-Flow MaX component has four core parts:

- The *constraint and rule definition* part is used to specify quality constraints for service calls and exception handling rules for events occurring during process execution. The specification has to be done manually, e.g. during process definition.
- The *constraint and event monitoring* checks whether called services violate any quality constraint or whether any other event (e.g., a fault message or a specific database update) occurs.
- The *exception handling* part uses the specified rules to determine whether an event constitutes an exception and how the exception should be handled. The main goal of the exception handling is to successfully continue the cooperative process so that quality constraints are met to the largest extent possible. This may require actions such as execution of additional services or adapting the process to compensate for the effects of the exception.
- A *logging* component records all events and exceptions together with the performed handling. The goal is to use this data for process optimization, in particular to provide recommendations for a manual exception handling and to eventually provide input for defining new exception handling rules.

Further components of the architecture are the *web service search* to use service directories for finding appropriate providers for a task to be executed, and the *Web-Flow metadata* repository maintaining metadata such as constraints and rules for quality monitoring and exception handling. Information about the cooperative processes and

the used services can be derived from the process database.

The Web-Flow MaX component uses a multilevel approach for exception handling. First, exception handling may take place at the site where a web service is executed, either by the web service itself or by the Web-Flow MaX component of the partner. For instance, a web service searching for offers in a particular price range may check the offers before returning them and perform a new search if no suitable results are found. Such mechanisms are typically invisible for the service user but require a sophisticated web service which cannot always be assumed. Therefore, additional levels of exception handling take place in the cooperative process and by the Web-Flow MaX component of the site invoking a remote web service. The Web-Flow MaX exception handling is largely independent of the respective web service implementations (i.e., can also be used for legacy applications wrapped as web services) and process specifications and thus enables enforcement of general exception handling policies.

A rule-based approach has the advantage that the standard process definition is clearly separated from the exception rules, therefore facilitating the readability and maintenance of both. Furthermore, necessary actions are not derived until an exception really occurs, so changes in business policies, laws, etc. can be taken into account without changes in the process definition.

### 3 Quality Constraints

Based on an analysis of different cooperative business process scenarios and on previous constraint classifications (e.g., [2]) we have identified several types of web-service-related quality constraints to be supported in Web-Flow:

- *Metadata constraints* refer to conditions on the description of web services, e.g. on the services' UDDI (Universal Description, Discovery and Integration, [14]) or WSDL metadata. Such constraints can restrict many different aspects, e.g. the provider of a service (specific companies, specific geographic locations,...) or a fee that may have to be paid for using a service (cost constraint).
- *Execution constraints* refer to conditions on the physical execution of web services, in particular response time limits or maximal number of re-trials for failed executions. Response time constraints specify the maximal waiting time for an answer from a service either as a fixed point in time (date) or as a time interval.
- *Input constraints* pose conditions on the input parameters of a web service call.
- *Result constraints* specify conditions on the result of a web service execution. These conditions typically refer to the XML documents returned by a service. They can be used to check whether the delivery time or price for a product is in an acceptable range or whether a returned product configuration matches the user requirements.

Metadata constraints refer to rather *static* information, which is specified when a service is registered. These constraints are primarily useful for *service selection*, i.e. to find a suitable service during process execution or after an exception has occurred. The other types of constraints are more *dynamic* in the sense that they refer to information related to the actual execution of web services (not only specification). They are of primary rel-

```

<service name="HotelSearchService">
  <operation operationName="hotelReservation">
    <resultConstraint name="RC1"
      scope="HotelSearchService:hotelReservation"
      strictness="high">
      <parameterCondition>
        <lessEqual>
          <leftOperand>
            <xPathQuery>/hotelDetails/maximumPrice</XPathQuery>
            <parameter name="hotelReservationRequest" type="Input" />
          </leftOperand>
          <rightOperand>
            <xPathQuery> /reservation/reservationDetails/price</XPathQuery>
            <parameter name="hotelReservationResponse" type="Output" />
          </rightOperand>
        </lessEqual>
      </parameterCondition>
    </resultConstraint>
  </operation>
  ...
</service>

```

**Fig. 2.** Sample result constraint

evance for *dynamic exception handling*.

There are several additional properties which characterize quality constraints and their use:

- *Monitoring*: Dynamic quality constraints (response time, input, result constraints) may either be monitored at the *consumer side* or at the *provider side* of a service. The first alternative is typical for a simple cooperation model preserving a high autonomy of service providers. A complex cooperation model may support both approaches.
- *Strictness*: Constraints may be mandatory or only desirable (high vs. low strictness). A service violating a constraint of the latter type may thus still be useful.
- *Scope*: Quality constraints typically refer to a particular service or a service operation, but may also relate to a particular process or user. In addition, there may be *global constraints* applying to several or all services, e.g. to enforce company-wide regulations on eligible service providers / cooperation partners or to specify a default value for maximal response times. Moreover, a constraint may be *context-dependent*, i.e. it should only apply if a service is used in a process under certain conditions, e.g. depending on preceding service calls. For instance, in a medical context the allowed ranges for a blood value may depend on the drugs applied beforehand.

In Web-Flow, a declarative XML-based specification has been defined for each type of quality constraint. Constraints are specified as logical comparison predicates which can be combined to a more complex condition using boolean AND, OR and NOT operators. The XML fragment in Fig. 2 shows an operation-specific result constraint that specifies that the price for a reserved hotel room has to be less or equal than the maximal price specified by the user. To check this constraint the output document of the operation has

to be compared to the input document previously used to call the web service. Therefore the data of the input document has to be stored until the service returns a response and the constraint can be evaluated. The *xPathQuery* tag specifies which part of the document specified by the *parameter* tag has to be used for evaluation.

All quality constraints are defined in the constraint and rule definition component of the Web-Flow MaX and are maintained in the Web-Flow metadata repository. Service and operation specific quality constraints are saved in the Web-Flow extended service description (\*.wesd). Process specific constraints are part of the Web-Flow extended process description (\*.wepd) of a cooperative process. Global constraints are maintained separately.

## 4 Dynamic Exception Handling

Dynamic exception handling in Web-Flow is based on a rule-based approach using Event-Condition-Action (ECA) rules (as e.g. used in active database systems [15]). Events that may result in an exception include time-outs, web service calls, reception of web service output or fault messages, or manual notifications by e-mail or phone call. Furthermore, database updates (e.g., in a database with customer data or appointments) can also be events. Events are detected by the constraint and event monitoring of the Web-Flow MaX component. It observes all messages that are sent or received by the process engine during process execution.

The optional condition part of a rule is used to check the various quality constraints and to specify additional conditions, e.g. on Web-Flow logging information or metadata, that need to hold so that an event results in an exception. This can be useful to express context-dependent constraints. We use a query processor of a database engine to check metadata, input and result constraints. The violation of a response time constraint can be detected by the monitoring component if no response is received for a synchronous service call until the deadline has expired. An execution constraint specifying a maximal number of re-trials is checked using Web-Flow logging information.

The action part specifies how an exception should be handled. Possibilities include delegating exception handling to the respective process, manual reaction, search for an alternative web service, invocation of a particular web service, process abort, and dynamic adaptation of the calling process, e.g. by adding/deleting activities. In Web-Flow, a default rule is created for each dynamic quality constraint specifying that a user has to be notified if the constraint is violated. These rules can be manually edited to specify alternative automatic reactions.

Fig. 3 shows an example rule specifying that an alternative service operation (operation B of service 2) should be called if the result constraint named RC1 (Fig. 2) is violated and the service has already been called two times. Rule priorities (PRIORITY part) are used to determine the order of rule execution if several rules apply per event (1 = highest priority).

Event processing includes logging the event and checking all relevant rules. This is performed as soon as an event is detected by the constraint and event monitoring component of the Web-Flow MaX. If a rule is found for an event specifying an automatically executable action Web-Flow will execute this action. If execution fails a manual

<i>EVENT</i>	<i>reception of output message o</i>
<i>CONDITION</i>	<i>violation of constraint RCI AND # iteration &gt; 2</i>
<i>ACTION</i>	<i>callService(service2.B)</i>
<i>PRIORITY</i>	<i>3</i>

**Fig. 3.** Sample exception handling rule

reaction is needed.

To reduce the dependence on manual exception handling we intend to evaluate all logged successful exception handling steps. When an exception occurs asking for a manual reaction (e.g. due to the default rule for quality constraints), the logged information should be searched to present the actions taken in similar exception cases in the past to the user. One difficulty with this step is to find appropriate metrics for similarity of exceptions, e.g. taking into account the service called, parameter values etc.

## 5 Conclusion and Future Work

To support a high execution quality of web-service-based cooperative processes dynamic handling of exceptions is needed. The Web-Flow approach presented in this paper is based on the specification of a variety of quality constraints for heterogeneous and autonomous web services. A rule-based monitoring and exception handling component automatically deals with many exceptions such as the violation of quality constraints or service faults.

We are currently implementing the monitoring, exception handling and logging algorithms of the Web-Flow MaX component. We intend to evaluate the approach not only in cross-organizational processes in the e-business domain but also in distributed health care scenarios.

## Acknowledgements

This work has been supported by the German Research Association under grant Ra 497-12.

## References

1. Chappell, D.A., Jewell, T.: Java Web Services. O'Reilly & Associates, Inc. (2002)
2. Keidl, M., Seltzsa, S., Kemper, A.: Reliable Web Service Execution and Deployment in Dynamic Environments. In: Benatallah, B., Shan, M-C. (eds.): TES 2003. Springer-Verlag, Berlin Heidelberg (2003) 104-118
3. Schuler, C., Weber, R., Schuldt, H., Schek, H.J.: Peer-to-Peer Process Execution with Osiris. In: Proc. of First International Conference on Service-Oriented Computing ICSOC (2003)
4. Keller, A., Ludwig, H.: The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services. J. Electr. Commerce Research **3** (1&2) (2003)
5. DAML-S Coalition: DAML-S: Web Service Description for the Semantic Web. In: Proc. of the First International Semantic Web Conference (ISWC), Italy, June 2002 (2002)

6. van der Aalst, W.M.P., Weske, M.: The P2P Approach to Interorganizational Workflows. In: Proc. of CAiSE 2001 (2001) 140-156
7. Grefen, P., Aberer, K., Ludwig, H., Hoffner, Y.: CrossFlow: Cross-Organizational Workflow Management for Service Outsourcing in Dynamic Virtual Enterprises. In: IEEE Bull. of the Tech. Comm. on Data Engineering **24** (1) (2001) 52-57
8. Reichert, M., Bauer, T., Fries, T., Dadam, P.: Realisierung flexibler, unternehmensweiter Workflow-Anwendungen mit ADEPT. In: Horster P. (ed.): Proc. Elektr. Geschäftsprozesse - Grundlagen, Sicherheitsaspekte, Realisierungen, Anwendungen. Klagenfurt (2001) 217-228
9. Luo, Z., Sheth, A., Kochut, K., Arpinar, B.: Exception Handling for Conflict Resolution in Cross-Organizational Workflows. J. of Distributed and Parallel Databases **13** (2003) 271-306
10. Casati, F., Shan, M.-C.: Dynamic and adaptive composition of e-services. Inf. Systems **26** (2001) 143-163
11. Chiu, D.K.W., Cheung, S.-C., Karlapalem, K., Li, Q., Till, S.: Workflow View Driven Cross-Organizational Interoperability in a Web-Service Environment. In: Proc. of Web Services, E-Business, and the Semantic Web, CAiSE 2002 International Workshop, WES 2002, Canada (2002) 41-56
12. Andrews, T. et al.: Business Process Execution Language for Web Services, Version 1.1, May 2003. <http://www-106.ibm.com/developerworks/library/ws-bpel/>
13. Christensen, E. et al.: Web Services Description Language (WSDL) 1.1. <http://www.w3.org/TR/wsdl> (2001)
14. Bellwood, T. et al.: UDDI Version 3.0, July 2002. <http://www.oasis-open.org/committees/uddi-spec/doc/tcspecs.htm> (2002)
15. Paton, N. (ed.): Active Rules in Database Systems. Springer-Verlag, New York (1999)

# Funktionsgetriebene Integration von Legacy-Systemen mit Web Services

T. Teschke<sup>1</sup>, H. Jaekel<sup>1</sup>,  
S. Krieghoff<sup>2</sup>, M. Langnickel<sup>2</sup>,  
W. Hasselbring<sup>3</sup> und R. Reussner<sup>3</sup>

<sup>1</sup> OFFIS, Bereich „Betriebliches Informations- und Wissensmanagement“

{teschke|jaekel}@offis.de

<sup>2</sup> Kommunale Datenverarbeitung Oldenburg (KDO)

{krieghoff|langnickel}@kdo.de

<sup>3</sup> Universität Oldenburg, Department für Informatik,  
Abteilung Software Engineering

{hasselbring|reussner}@informatik.uni-oldenburg.de

**Zusammenfassung** Dieser Beitrag beschreibt die Nutzung von Web Services bei der Migration eines Legacy-Systems. Es wird eine Architektur vorgestellt, auf deren Grundlage die *Kommunale Datenverarbeitung Oldenburg (KDO)* ihre umfangreichen Legacy-Anwendungssysteme in eine moderne Softwarearchitektur integriert. Grundlage dieser Architektur ist das Dublo-Architekturmuster, das die sanfte Integration und Migration von Legacy-Anwendungssystemen unterstützt. Das betrachtete kommunale Anwendungssystem wird über Web Services in eine J2EE-Architektur integriert, wobei die zustandsabhängige Verfügbarkeit einzelner Funktionen anhand von Protokollautomaten kontrolliert wird.

## 1 Einleitung

Die *Kommunale Datenverarbeitung Oldenburg (KDO)* ist ein Softwarehaus und IT-Dienstleister für die kommunale Verwaltung. Angesichts des enormen Kostendrucks im kommunalen Bereich und wachsender bzw. sich ändernder Aufgaben der Verwaltungen müssen die von der KDO entwickelten kommunalen Informationssysteme flexibel und kostengünstig an neue, kundenspezifische Anforderungen anpassbar und somit langfristig nutzbar sein.

Bislang sind die Client/Server-Informationssysteme der KDO auf der Grundlage von Informix-Datenbanksystemen und der 4Js-Implementierung [Fou03] der Informix 4GL [IBM03] entwickelt worden, wobei Präsentations- und Anwendungscode in den 4GL-Programmen eng verzahnt sind. 4GL ist eine so genannte „fourth generation language“ [WW90]. Maßnahmen zur Sicherung der Konsistenz der in den Datenbankrelationen abgelegten Daten sind nicht auf Datenbankebene, sondern durch die Fachverfahren realisiert. Dies ist durch die Notwendigkeit begründet, den Betrieb auch mit alten Datenbankversionen sicherzustellen, da Kommunen aus Kostengründen nicht regelmäßig in neue Datenbanktechnologie investieren bzw. Vollwartung abschließen können.

Die KDO hat sich dazu entschlossen, diese (monolithische) 2-Schichten-Architektur mittelfristig durch eine flexiblere Mehrschichtenarchitektur zur ersetzen, die auf aktuellen Standards aufsetzt und moderne Techniken des Software Engineering berücksichtigt. Da die aktuelle Implementierung der Fachverfahren umfangreiches Domänenwissen beinhaltet und daher nicht einfach ersetzt werden kann, sollen diese Altverfahren zunächst über Web Services in die neue Architektur eingebunden und später schrittweise abgelöst werden. Das Oldenburger Forschungs- und Entwicklungsinstitut für Informatik-Werkzeuge und -Systeme (OFFIS), ein An-Institut der Carl von Ossietzky Universität Oldenburg, begleitet die KDO bei dieser Umstellung ihrer Softwarearchitektur. Die in diesem Beitrag vorgestellte Arbeit ist das Ergebnis einer Kooperation des OFFIS-Bereichs „Betriebliches Informations- und Wissensmanagement“, der Abteilung „Software Engineering“ des Departments für Informatik und der KDO.

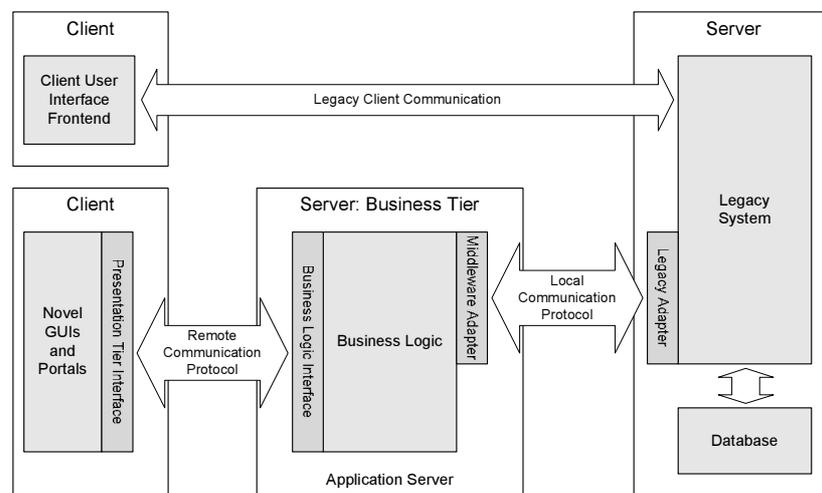
Dieser Beitrag ist wie folgt aufgebaut: Ausgehend von einer kurzen Vorstellung des Dublo-Architekturmusters in Abschnitt 2 skizzieren wir in Abschnitt 3 drei grundsätzliche Alternativen zur Identifikation von Web Services in Legacy-Systemen. In Abschnitt 4 beschreiben wir den bei der KDO verfolgten Ansatz zur Identifikation von Web Services in den Altverfahren und deren Integration in die neue Architektur, bevor wir den Beitrag in Abschnitt 5 mit einer Zusammenfassung und einem Ausblick beschließen.

## 2 Sanfte Legacy-Integration und -Migration nach dem Dublo-Muster

Legacy-Anwendungssysteme unterscheiden sich von modernen Softwarearchitekturen oft darin, dass sie monolithisch aufgebaut sind und nicht zwischen unterschiedlichen Schichten für Präsentation, Anwendungslogik und Datenhaltung unterscheiden. Stand der Softwaretechnik sind heute jedoch mehrschichtige Architekturen mit klarer Trennung von Präsentation, Anwendungslogik und Datenhaltung, mit denen u. a. vereinfachte Austauschbarkeit und Anpassbarkeit von Komponenten, verbesserte Nachvollziehbarkeit fachlicher Anforderungen in der Schicht der Geschäftslogik oder erhöhte Transparenz von Datenbankaspekten angestrebt werden.

Eine sanfte Migration von monolithischen Legacy-Systemen zu modernen, mehrschichtigen Softwarearchitekturen gestattet das *Dublo-Muster* [HRJ<sup>+</sup>04]. Dublo steht für „Dual Business Logic“ und basiert im Kern auf dem Gedanken, Geschäftslogik an zwei Stellen zu halten: im Legacy-System sowie in neu eingeführten Mittelschichten. Dabei wird (neue) Geschäftslogik in einer neuen Geschäftslogikschicht implementiert und mittels eines Legacy-Adapters [GHJV96] mit der alten Geschäftslogik verbunden. Neben dem Zugriff auf alte Geschäftslogik wird dieser Adapter insbesondere auch für den Datenbankzugriff genutzt, d. h. Zugriffe der neu implementierten Geschäftslogik auf die Datenbank erfolgen ausschließlich über den existierenden Legacy-Code. Dieser Zugriff auf die Datenbank durch bestehenden Legacy-Code führt zwar zu einer Verteilung von Geschäftslogik auf zwei Schichten, hat aber den Vorteil, dass alte Geschäftslogik

zunächst weiter verwendet werden kann (um schließlich schrittweise ersetzt zu werden). Insbesondere kann so aufwändige Geschäftslogik des Legacy-Systems, die (wie bei 4GL Systemen häufig) intrinsisch mit dem Datenbenkzugriff ver-  
woben ist, weiter genutzt werden und mehr oder weniger unverändert durch die  
mittleren Schichten zu einer neuen Präsentationsschicht weitergereicht werden.  
Im Laufe der Migration wird dann immer mehr Geschäftslogik portiert, bis der  
Legacy-Adapter schließlich vollständig durch einen Datenbank-Adapter ersetzt  
werden kann. Abbildung 1 zeigt die Struktur des Dublo-Architekturmusters.



**Abbildung 1.** Strukturelle Sicht auf das Dublo-Architekturmuster

Im Kontext der KDO wird das Dublo-Architekturmuster eingesetzt, um einen J2EE-Server [Sha01] an das bestehende Informix-4GL-System anzubinden. Dabei werden die Legacy-Adapter über Web Services realisiert. Von der Nutzung einer standardisierten, plattformunabhängigen Technologie wie Web Services verspricht man sich insbesondere Flexibilität bei Technologiewechseln in der Datenbank- und Geschäftslogikschicht [PG03]. Alternativen zur Identifikation geeigneter Web Services werden im folgenden Abschnitt diskutiert. Die KDO wird die durch das Dublo-Muster eingeführte neue Präsentationsschicht ergänzend zum Zugriff auf neue Geschäftslogik auch für den Zugriff auf alte Geschäftslogik entwickeln, um die alte Präsentationsschicht abzulösen.

Die Verbindung zwischen Präsentationsschicht und Geschäftslogikschicht muss aus Gründen der Datensicherheit gesondert gesichert werden. Dazu kann in einer Java-Umgebung beispielsweise RMI und SSL eingesetzt werden. Im Kontext der KDO wurde diese Verbindung mittels OSCI (Online Services Computer Interface) [OSC] gemäß gesetzlichen Vorgaben gesichert. Durch Sicherung der

Verbindung zwischen Präsentations- und Geschäftslogikschicht laufen Server, die diese Schichten realisieren, in einer gesicherten Umgebung, d. h. insbesondere kann die Verbindung zwischen Geschäftslogik- und Datenbankschicht als gesichert angesehen werden.

### 3 Alternativen zur Identifikation von Web Services

Bei der Identifikation von Web Services (oder allgemeiner: Klassen) in Legacy-Systemen lassen sich nach [WBF97] drei alternative Ansätze unterscheiden:

1. *Datengetrieben*: Identifikation von Web Services zur Repräsentation des Objektmodells auf der Grundlage der im Legacy-System implementierten Datenstrukturen.
2. *Funktionsgetrieben*: Identifikation von Web Services zur Bündelung der für die Durchführung zentraler Anwendungsfälle relevanten Dienste auf der Grundlage der im Legacy-System realisierten Funktionalität.
3. *Objektgetrieben*: Erstellung eines Objektmodells der Anwendungsdomäne und Analyse des Legacy-Systems, um Code-Elemente zu identifizieren, die die Datenstrukturen und Funktionalität entsprechender Klassen implementieren.

Im Kontext der KDO erschien die objektgetriebene Identifikation von Web Services, die z. B. auf einem durch Entity Beans definierten Ziel-Objektmodell aufsetzen könnte, grundsätzlich nicht gangbar. Da die den Altverfahren zugrunde liegenden Relationen im Allgemeinen nicht normalisiert sind und sich daraus erhöhte Differenzen zwischen Ist- und (normalisiertem) Ziel-Objektmodell ableiten ließen, waren bei diesem Ansatz Schwierigkeiten bei der Zuordnung von Funktionen der Fachverfahren zu den Entity Beans des Ziel-Objektmodells zu erwarten.

Ähnlich wie der objektgetriebene Ansatz zielt auch die datengetriebene Identifikation von Web Services auf die Implementierung eines Objektmodells durch Legacy-Funktionen ab. Der wesentliche Unterschied zwischen beiden Ansätzen ist darin zu sehen, dass der objektgetriebene Ansatz von einem Ziel-Objektmodell ausgeht, während der datengetriebene Ansatz auf dem aktuell implementierten Objektmodell aufbaut. Beim datengetriebenen Ansatz werden geeignete Web Services im Sinne abstrakter Datentypen (ADTs) ausgehend von den im Legacy-System implementierten Datenstrukturen identifiziert. Dies können zum einen (komplexe) Datentypen, die im Quellcode der Anwendung genutzt werden, und zum anderen (relationale) Datenbankschemata sein. In der Forschung sind verschiedenartige Ansätze auf Basis von Techniken wie Clustering und Konzeptanalyse (vgl. z. B. [vDK99]) oder Datenfluss-, Kontrollfluss- und Syntexanalyse (vgl. z. B. [KP99]) untersucht worden.

Der funktionsgetriebene Ansatz zielt im Kern auf die „Entdeckung“ von Anwendungsfällen in Legacy-Systemen ab, für die durch geeignet zusammengestellte Web Services eine Schnittstelle definiert werden soll. Dabei kann sich diese Ermittlung von Anwendungsfällen an den einzelnen Legacy-Programmen, die

die betriebliche Funktionalität des Legacy-Systems implementieren, oder an den Bildschirmmasken des Systems orientieren. Erstgenannter Ansatz wird z. B. von ERLIKH [Erl02] verfolgt, der zunächst Web Services durch iterative Verfeinerung von Mengen funktional zusammengehöriger Programme identifiziert. Die APIs dieser Web Services werden anschließend durch Einsatz verschiedener Techniken des *Knowledge Mining* ermittelt. Bei dem maskenorientierten Ansatz, den STROULIA ET AL. in [SERS02] vorgestellt, werden durch die Analyse der Benutzerinteraktionen mit dem Legacy-System Anwendungsfälle extrahiert, für die eine neue Schnittstelle (in unserem Fall in Form von Web Services) anzubieten ist. Durch die Bestimmung der für die Durchführung dieser Anwendungsfälle genutzten Funktionen lassen sich die APIs entsprechender Web Services definieren.

#### 4 Identifikation und Nutzung von Web Services bei der KDO

Da sich die Struktur der Implementierung der einzelnen kommunalen Fachverfahren und ihrer Funktionen primär an den unterstützten Anwendungsfällen und Geschäftsprozessen und weniger an den zugrunde liegenden Datenbankstrukturen orientiert, hat sich die KDO für einen funktionsgetriebenen Ansatz zur Identifikation und Nutzung von Web Services entschieden. Dabei erschien das von ERLIKH [Erl02] vorgeschlagene Verfahren aufgrund des Einsatzes von Techniken des Knowledge Mining zu komplex. Wesentliche Gedanken des von STROULIA ET AL. [SERS02] untersuchten Ansatzes ließen sich dagegen auf die KDO übertragen, da in den Fachverfahren eine enge Kopplung zwischen den Bildschirmmasken und den Funktionen besteht. Sein Einsatz setzt allerdings voraus, dass die Struktur der aktuellen Masken in der J2EE-Implementierung im Wesentlichen beibehalten bleibt.

Die Identifikation der Web Services, die zur Nutzung der Altverfahren zu exportieren sind, orientiert sich an den bestehenden Fachverfahren und deren Bildschirmmasken und wird manuell durchgeführt. Für jede Maske werden dabei die Benutzerinteraktionen ermittelt, mit denen datenverarbeitende Vorgänge ausgelöst werden, die jeweils auszuführenden Funktionen analysiert und (für jede dieser Interaktionen) entsprechende Operationen eines Web Services definiert. Ein Web Service kann dabei die Operationen einer Maske oder einer funktional zusammenhängenden Sequenz von Masken umfassen. Die Parameter der Operationen eines Web Service werden aus den Feldern der jeweiligen Eingabemaske abgeleitet. Die für einen Web Service zulässigen Interaktionsfolgen (Sequenzen von Operationsaufrufen) werden durch Analyse der Abhängigkeiten zwischen Bildschirmmasken bestimmt und in Form eines endlichen (Protokoll-) Automaten spezifiziert, um die korrekte Nutzung der Dienste der Altverfahren sicherzustellen. Sind Bildschirmmasken nicht vorhanden, oder sind ihre Abhängigkeiten nicht (implizit oder explizit) spezifiziert, so können auch Verfahren zur Analyse von Geschäftsprozessen („Process Mining“) genutzt werden, um die Menge zulässiger Interaktionsfolgen zu bestimmen [Sch02]. Die explizite Angabe von zulässigen Sequenzen ermöglicht die Erkennung von unzulässigen Aufrufe-

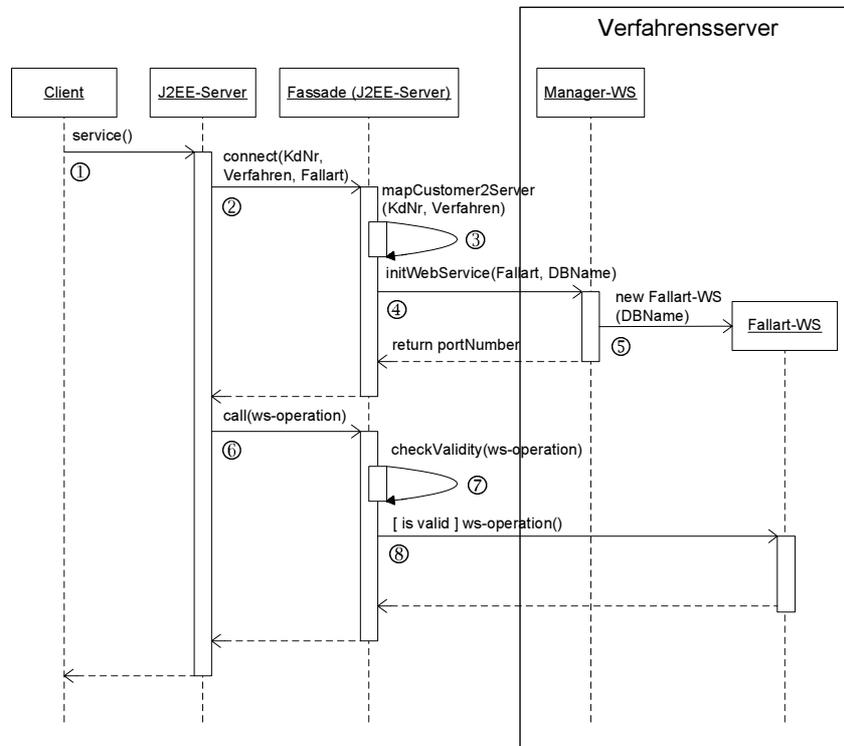
quenzen zur Laufzeit. Der Vorteil dieser Erkennung liegt in der exakten Identifikation des ersten unzulässigen Aufrufs. Würde man diese Überprüfung unzulässiger Aufrufreihenfolgen nicht durchführen, würde das Fehlverhalten, welches durch eine unzulässige Aufrufreihenfolge verursacht wird, entweder gar nicht oder aber möglicherweise so spät entdeckt, dass eine Identifikation des unzulässigen Aufrufs nur schwer oder gar nicht möglich ist. Beschreibt man nicht nur die zulässigen Aufrufsequenzen des Adapters, sondern zusätzlich auch die von der Geschäftslogikschicht benötigten Dienste, ist eine statische Interoperabilitätsprüfung der Aufrufsequenzen (Protokolle) möglich. Wird diese Prüfung schon zur Entwicklungszeit durchgeführt, können Überprüfungen der Aufrufsequenzen zur Laufzeit wegfallen, da eine statische Prüfung alle inkompatiblen Interaktionsfolgen ausschließt. Da der Aufwand einer Modellierung der von der Geschäftslogik benötigten Aufrufsequenzen als zu hoch eingeschätzt wurde, verzichtete man im beschriebenen Projekt auf statische Überprüfungen.

Ein Ansatz, der mit unserer Automatenutzung vergleichbar ist, wird bei verschiedenen Beschreibungssprachen für Web Services verfolgt. So lässt sich im Web Service Choreography Interface (WSCI) [W3C02] das beobachtbare Verhalten eines Dienstes als temporale und logische Abhängigkeiten zwischen den ausgetauschten Nachrichten beschreiben. Die Business Process Execution Language for Web Services (BPEL4WS) [BEA03] definiert eine Notation, die die temporalen und logischen Bedingungen bei der Zusammensetzung von Geschäftsprozessen aus mehreren Web Services beschreibt. Formal lassen sich Sequenzen von Web Services mit den gängigen Notationen zur Protokollspezifikation, also z.B. endlichen Automaten oder Petri-Netzen, beschreiben [BRSM03, MPP02]. Da noch nicht absehbar ist, welcher der konkurrierenden Standards WSCI oder BPEL4WS sich zukünftig durchsetzen wird, werden bei der KDO endliche Automaten zur Beschreibung der zulässigen Interaktionsfolgen eingesetzt.

Eine Voraussetzung für den Zugang zur implementierten Funktionalität über Web Services ist die Trennung der GUI-Aspekte von dem eigentlichen Anwendungscode durch Auslagerung des jeweiligen Codes in eigenständige 4GL-Funktionen. Dabei werden Interaktionen, die bislang aufgrund der Verzahnung von GUI- und Anwendungsaspekten innerhalb einer Funktion abgefragt und verarbeitet wurden, durch die Aufspaltung der 4GL-Funktionen an den Interaktionspunkten berücksichtigt. Die Web-Service-Schnittstelle kann dann auf technischer Ebene als Adapter für die resultierenden 4GL-Anwendungsfunktionen bereitgestellt werden.

Der Zugriff auf die Altverfahren der KDO über Web Services kann gemäß dem in Abbildung 2 dargestellten Sequenzdiagramm erfolgen. Dabei werden die folgenden Schritte ausgeführt:

- ① Aufruf eines Dienstes durch einen (Präsentations-)Client: Ein Client ruft eine Methode einer zustandsbehafteten Session Bean [DYK01] auf, die durch einen J2EE-Server bereitgestellt wird.
- ② Initialisierung der Session Bean: Sofern die Session Bean noch nicht initialisiert wurde, meldet diese sich zunächst bei einem „Fassaden-Server“ mit einer Kundenkennung sowie dem gewünschten Verfahren und der auszuführenden



**Abbildung 2.** Integration von Altverfahren über Web Services

Fallart an (der Fassaden-Server bietet in Anlehnung an das gleichnamige Entwurfsmuster [GHJV96] eine abstrakte Schnittstelle zu den Altverfahren an). Ist die Session Bean bereits initialisiert, wird dieser Schritt übersprungen und direkt mit Schritt 6 fortgefahren.

- ③ Auswahl des Verfahrensservers: Der Fassaden-Server, dessen Dienste ebenfalls über (zustandsbehaftete) Session Beans implementiert sein können, wählt anhand der Kundenkennung und des Verfahrens einen zuständigen Verfahrensserver aus. Dabei erfolgt auch die Bestimmung der durch das Verfahren zu benutzenden Datenbank.
- ④ Beantragung einer Web-Service-Instanz: Der Fassaden-Server beantragt bei einem „Manager-Web-Service“, der auf dem zuvor ermittelten Verfahrensserver läuft und die Erzeugung von Web Services kontrolliert, eine Instanz des für die gewünschte Fallart zuständigen Web Services und übergibt dabei den Namen der zu benutzenden Datenbank.
- ⑤ Erzeugung der Web-Service-Instanz: Der Manager-Web-Service legt eine Instanz des gewünschten Web Services als neuen Prozess an (da die 4Js-Programme der Altverfahren globale Variablen nutzen, erfolgt hier eine Tren-

nung zwischen unterschiedlichen Clients durch die Erzeugung unabhängiger Prozesse). Er gibt die Nummer des Ports zurück, über den der Prozess identifiziert und angesprochen werden kann. Mit der anschließenden Speicherung dieser Port-Nummer auf dem Fassaden-Server wird die Initialisierung der Session Bean (Schritt ②) abgeschlossen.

- ⑥ Anforderung des Dienstes: Die zustandsbehaftete Session Bean fordert die dem clientseitig angeforderten Dienst entsprechende Operation des Web Services beim Fassaden-Server an.
- ⑦ Prüfung der Gültigkeit des Aufrufs: Der Fassaden-Server prüft die Zulässigkeit des angeforderten Aufrufs anhand eines Protokollautomaten, der die zulässigen Interaktionsfolgen einer Web-Service-Session beschreibt.
- ⑧ Ausführung der Operation: Ist der angeforderte Aufruf zulässig, wird die Operation des Web Services ausgeführt und Rückgabewerte an den Client zurückgegeben, anderenfalls wird ein Fehler signalisiert.

Aus Platzgründen verzichten wir hier auf eine detaillierte Betrachtung des Abbaus von Verbindungen zwischen J2EE-Server und Web-Service-Instanz. Inwieweit die zusätzliche Serverlast, die durch die Erzeugung einer möglicherweise hohen Anzahl von Web-Service-Prozessen entsteht, kritisch ist, muss sich im praktischem Einsatz noch zeigen.

## 5 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Ansatz zur funktionsgetriebenen Integration von Legacy-Systemen mit Web Services im Kontext kommunaler Informationssysteme vorgestellt, den wir als Weiterentwicklung bestehender Diskussionen auf dem Gebiet der Anwendungsintegration einordnen.

Im Anschluss an eine kurze Einleitung, in der die Ausgangssituation der KDO skizziert wurde, folgte eine knappe Einführung in die sanfte Integration und Migration von Legacy-Anwendungssystemen nach dem Dublo-Architekturmuster. Eine Analyse grundsätzlicher Alternativen zur Identifikation von Web Services in Legacy-Systemen bildete die Grundlage für den nachfolgend vorgestellten Ansatz zur Identifikation und Nutzung von Web Services, über den die umfangreichen kommunalen Fachverfahren in die moderne Mehrschichtenarchitektur der KDO eingebunden werden. Um die Zulässigkeit der Reihenfolge von Web-Service-Aufrufen sicherzustellen, kommen in der vorgestellten Architektur Spezifikationen endlicher Automaten zum Einsatz, auf deren Grundlage zulässige Interaktionsfolgen mit dem Legacy-System beschrieben werden.

Da die Verwendung des Dublo-Musters aufgrund der auf zwei Schichten verteilten, z. T. duplizierten Geschäftslogik einen gewissen Mehraufwand für die Pflege des Anwendungssystems impliziert, beschränkt sich die Anwendbarkeit dieses Musters auf „stabile“ Legacy-Systeme, die keine großen funktionalen oder strukturellen Änderungen mehr erfahren. Als Schwachpunkt des in diesem Beitrag vorgestellten Ansatzes muss die Beibehaltung des alten, oftmals im Laufe

der Jahre degenerierten Datenbankschemas in neuen Fachanwendungen betrachtet werden. Künftige Arbeiten betreffen daher die Konzipierung eines Migrationspfades zur vollständigen Ablösung von Legacy-Systemen und die dabei durchzuführende Evolution des Datenbankschemas. Während des parallelen Betriebs alter und neuer Geschäftslogik ist zu erwarten, dass die semantische Heterogenität zwischen alter und neuer Geschäftslogik noch weitere Herausforderungen birgt.

## Literatur

- [BEA03] BEA, IBM, MICROSOFT, SAP AG UND SIEBEL SYSTEMS, <http://www.ibm.com/developerworks/library/ws-bpel/>: *Business Process Execution Language for Web Services Version 1.1*, 2003. Besucht am 18.12.2003.
- [BRSM03] BERARDI, D., F. DE ROSA, L. DE SANTIS und M. MECELLA: *Finite State Automata as Conceptual Model for E-Services*. In: *Integrated Design & Process Technology*. Society for Process & Design Sciences, 2003.
- [DYK01] DEMICHEL, L. G., L. Ü. YALÇINALP und S. KRISHNAN: *Enterprise JavaBeans Specification, Version 2.0*. Sun Microsystems, 2001.
- [Erl02] ERLIKH, L.: *Integrating Legacy Systems Using Web Services*. eAI Journal, 2002.
- [Fou03] FOUR J's, <http://www.4js.com>: *Four J's Development Tools*, 2003. Besucht am 25.08.2003.
- [GHJV96] GAMMA, E., R. HELM, R. JOHNSON und J. VLISSIDES: *Entwurfsmuster: Elemente wiederverwendbarer objektorientierter Software*. Addison-Wesley, 1. Auflage, 1996.
- [HRJ<sup>+</sup>04] HASSELBRING, W., R. REUSSNER, H. JAEKEL, J. SCHLEGELMILCH, T. TESCHKE und S. KRIEGHOFF: *The Dublo Architecture Pattern for Smooth Migration of Business Information Systems*. In: *26th International Conference on Software Engineering (ICSE) 2004*, Edinburgh, Schottland, Großbritannien, Mai 2004.
- [IBM03] IBM, <http://www-3.ibm.com/software/data/informix/tools/4gl/>: *Informix 4GL product family*, 2003. Besucht am 25.08.2003.
- [KP99] KONTOGIANNIS, K. und P. PATIL: *Evidence Driven Object Identification in Procedural Code*. In: *Proceedings of IEEE Conference on Software Technology and Engineering Practice (STEP '99)*, Seiten 12–21, Pittsburgh, Pennsylvania, USA, August 1999.
- [MPP02] MECELLA, MASSIMO, FRANCESCO PARISI PRESICCE und BARBARA PERINICI: *Modeling E-service Orchestration through Petri Nets*. Lecture Notes in Computer Science, 2444:38ff., 2002.
- [OSC] OSCI, <http://www.osci.de>: *Online Services Computer Interface*. Besucht am 18.12.2003.
- [PG03] PAPZOGLOU, M.P. und D. GEORGAKOPOULO: *Special Section on Service Oriented Computing*. Communications of the ACM, 46(10):24–61, Oktober 2003.
- [Sch02] SCHIMM, GUIDO: *Process Miner — A Tool for Mining Process Schemes from Event-Based Data*. Lecture Notes in Computer Science, 2424:525ff., 2002.

- [SERS02] STROULIA, E., M. EL-RAMLY und P. SORENSON: *From Legacy to Web through Interaction Modeling*. In: *Proceedings of the International Conference on Software Maintenance (ICSM '02)*, Seiten 320–329, Montreal, Kanada, Oktober 2002. IEEE Press.
- [Sha01] SHANNON, B.: *Java 2 Platform Enterprise Edition Specification, v1.3*. Sun Microsystems, 2001.
- [vDK99] DEURSEN, A. VAN und T. KUIPERS: *Identifying Objects Using Cluster and Concept Analysis*. In: *Proceedings of the 1999 International Conference on Software Engineering (ICSE '99)*, Seiten 246–255, Los Angeles, USA, Mai 1999. ACM.
- [W3C02] W3C, <http://www.w3.org/TR/wsci/>: *Web Service Choreography Interface (WSCI) 1.0*, 2002. Besucht am 18.12.2003.
- [WBF97] WIGGERTS, T., H. BOSMA und E. FIELT: *Scenarios for the Identification of Objects in Legacy Systems*. In: BAXTER, I. D., A. QUILICI und C. VERHOEF (Herausgeber): *Proceedings of the 4th Working Conference on Reverse Engineering (WCRE '97)*, Seiten 24–32, Amsterdam, Niederlande, Oktober 1997. IEEE Computer Society Press.
- [WW90] WOJTKOWSKI, W.G. und W. WOJTKOWSKI: *Applications Software Programming With Fourth-Generation Languages*. Wadsworth Publishing, 1990.

# Prozessorientierte B2B-P2P-Integration für Wissensmanagementanwendungen

Uwe Küssner

neofonie Technologieentwicklung und Informationsmanagement GmbH,  
Robert-Koch-Platz 4 , D-10115 Berlin,  
Uwe.Kuessner@neofonie.de

**Zusammenfassung** Workflowmanagementsysteme, die spezielle Unterstützung für interorganisationale Workflows bieten, sind ein vielversprechendes Integrationswerkzeug für B2B-Integrationen. Wir stellen Anforderungen an derartige Systeme vor und berücksichtigen dabei auch Anforderungen aus dem Bereich des Wissensmanagements. Wir präsentieren am Beispiel eines sich in der Entwicklung befindlichen Systems, wie die Anforderungen funktional realisiert und wie sie architektonisch auf Basis eines P2P-Protokolls umgesetzt werden und bewerten die eingesetzte Technologie JXTA.

## 1 Einführung

Die Integration von Unternehmensanwendungen(EAI) ist nicht nur eine informationstechnische Aufgabe. Aus Unternehmenssicht handelt es sich dabei in erster Linie um eine Maßnahme des Geschäftsprozessmanagement (GPM). Aus Sicht des GPMs geht es dabei darum, bestehende Prozesse zu optimieren oder zukünftige möglichst optimal zu gestalten, indem durch softwaretechnische Integration die Zusammenarbeit der Prozessschritte verbessert oder überhaupt erst ermöglicht wird. Insbesondere bei einer prozessorientierten Integration durch Workflowmanagementsysteme(WfMS) sollte vor der eigentlichen Integration eine Modellierung, Analyse und (Re)design der betreffenden Geschäftsprozesse(GP) durchgeführt werden. Auf der anderen Seite besteht ein enger Zusammenhang zwischen Wissensmanagement(WM) und GPM, siehe dazu z.B. [2]. Wissen und Prozesse sind eng miteinander verknüpft, da bei der Bearbeitung von Prozessschritten immer auch Wissen generiert, verteilt, bewahrt oder verwendet wird. Die Verarbeitung von Wissen ist immer kontextabhängig, der GP gibt diesen Kontext vor und ist somit gewissermaßen der „natürliche“ Ort zum Ansatz von WM. So gesehen ist es naheliegend, die Einführung eines prozessorientierten Integrationswerkzeugs (i.e. WfMS) nicht nur aus Sicht des GPMs, sondern integriert mit WM aufzusetzen. Bei der Analyse/Modellierung eines Workflows und Einführung eines WfMS sollten die Wissensaspekte dementsprechend mit berücksichtigt werden. Durch die zunehmende Virtualisierung von Unternehmen, zunehmendes Outsourcing von Prozessschritten und Durchdringung der Internettechnologie steigt der Bedarf an B2B-Integration. Eine technologische Antwort darauf sind B2B-WfMS, die die speziellen Anforderungen an

Standort- oder Unternehmensgrenzen überschreitende Workflows(Wfs) berücksichtigen. Auch aus WM-Sicht ergeben sich weitere Anforderungen an derartige B2B-WfMS gegenüber traditionellen WfMSen.

Ein interessanter technologischer Ansatz, derartige Systeme zu konstruieren, liegt mit dem Peer-to-Peer(P2P) basierten Kommunikationsmodell vor. Im Folgenden stellen wir die speziellen Anforderungen an WfMSe im B2B-Bereich vor und berücksichtigen dabei auch die Anforderungen des WMs. Schließlich stellen wir an Hand des konkreten Systems „neofonie architect:flow“ vor, wie die Anforderungen architektonisch und funktional umgesetzt werden und zeigen welchen Stellenwert dabei die verwendete P2P-Technologie hat. Das System befindet sich zur Zeit noch in der Entwicklung. Die Entwicklung ist teilweise mit BMBF-Mitteln gefördert.

## 2 B2B-Workflow-Management-System (B2B-WfMS)

WfMSe als Teil einer Integrationsplattform überwinden einige der Schwächen von Hub-and-Spoke-Architekturen nach dem Publish/Subscribe(P/S)-Kommunikationsmodell wie: Multistep-Integration, Geschäftslogik, bidirektionale Kommunikation (siehe [4], Seite 12). WfMSe, als Werkzeuge des GPMs, dienen der Automatisierung von GPMs. Voraussetzung zur Automatisierung ist die vorherige Erstellung eines formalen Modells (Wf-Schema), in dem der Workflow als Ablauf von einzelnen Aktivitäten beschrieben wird. Typische Modellelemente sind hier Sequenz, Splits (paralleler Ablauf), Joins (Synchronisationspunkte) und Bedingungen usw. Einen Vergleich der Ausdruckstärke von verschiedenen Systemen ist in [1] zu finden. Bei den Aktivitäten können automatische, von solchen mit manuellem Anteil unterschieden werden. Aus Sicht des WfMSs bestehen die Aktivitäten aus Programmen/Prozessen, die unter der Kontrolle des WfMSs ausgeführt werden. Gibt es einen manuellen Anteil, fungieren die Programme als Werkzeuge zur Bearbeitung des Prozessschrittes, die natürlich nicht von beliebigen, sondern nur von dafür vorgesehenen Mitarbeitern bedient werden sollen. In diesem Fall ist es Aufgabe des WfMSs die entsprechenden Aufgaben in die Taskliste der dafür vorgesehenen Mitarbeiter einzuordnen. Diese Zuordnung findet nicht zur Designphase des Wf-Schemas, sondern zur Laufzeit einer konkreten Wf-Instanz statt. Eine Wf-Instanz ist eine Repräsentation eines konkreten Wfs zu einem gegebenen Schema. Die Erzeugung von Wf-Instanzen gehört ebenfalls zu den Aufgaben des WfMSs. Herkömmliche WfMSe unterstützen nur Wfs, die innerhalb eines Unternehmens ablaufen, aber keine Standort übergreifenden oder gar Unternehmensgrenzen überschreitenden Wfs (B2B-Workflows). In diesem Bereich haben wir es mit besonderen Anforderungen zu tun:

**Sicherheit.** Im Vergleich zu intraorganisationalen Workflows gibt es bei interorganisationalen Workflows erhöhte Sicherheitsanforderungen:

Die *Authentizität* ist auf verschiedenen Granularitäten sicherzustellen. So muss verhindert werden, dass sich ein Unternehmen für ein anderes ausgeben

kann. Das gleiche gilt auf feinerer Granularität für Mitarbeiter, die einen Prozessschritt bearbeiten. Die Ergebnisse von einigen Prozessschritten müssen *vertraulich* behandelt werden und erfordern deswegen Verschlüsselung. Ein Sachbearbeiter, der eine Aktivität ausgeführt hat, darf dieses hinterher nicht erfolgreich abstreiten können (*Verbindlichkeit*). Unternehmensinterne Workflows sollen von außerhalb nicht einsehbar sein (*Verborgenheit*). Eine feingranulare Zugriffskontrolle muss sicherstellen, dass nur solche Mitarbeiter einzelne Prozessschritte und die damit verbundenen Dokumente bearbeiten können, die dafür zugeteilt sind.

**Autonomie, Architektur und Transaktionen.** Eine Reihe von Anforderungen leiten sich aus der Autonomie der partizipierenden Organisationen ab. Partizipanten gehen bezüglich der Teilnahme an einem B2-Wfs naturgemäß eine Verpflichtung ein, wollen aber darüberhinaus weitgehend autonom agieren können. Unternehmen wollen sich beispielsweise nicht von zentralen Komponenten außerhalb ihres Standorts abhängig machen, die wesentliche Teile ihrer Prozesse kontrollieren. Das gilt beispielsweise für Transaktionsmonitore, die Ressourcen sogar blockieren können. Bezüglich Transaktionen möchte man daher bei B2B-Workflows Alternativen zu ressourcenblockierenden Systemen haben. Daher sollen sog. Business-Transaktionen unterstützt werden, d.h. langandauernde Transaktionen unter Aufweichung der ACID-Bedingungen. Statt Isolation sollen z.B. Wf-spezifische Kompensationsaktivitäten möglich sein. Autonomie bedeutet auch, dass das Wf-Schema keine Annahmen bezüglich spezifischer Mitarbeiter, Anwendungen oder der Organisationsstruktur machen darf. Weiterhin sollte die konkrete Realisierung eines Workflows innerhalb der Organisation festgelegt werden können. Aus Sicht des Changemanagement bedeutet dies, dass die partizipierenden Organisationen ihre Struktur, Mitarbeiter und Anwendungen ändern können, ohne dass das für die anderen Partizipanten transparent wird. Aus WM-Sicht bedeutet dies, dass man nur soviel Wissen über Interna weitergibt, wie es die Zusammenarbeit erfordert.

**Wissensmanagement** Das System soll sowohl eine Personifizierungs- als auch eine Kodifizierungsstrategie [5] unterstützen. Bei der *Kodifizierungsstrategie* geht es vorrangig darum, Wissen zu externalisieren ([11]) und in kodifizierter Form (Dokumente etc) dem Anwender zur Verfügung zu stellen. Bei der *Personifizierungsstrategie* ist nicht das Generieren, Sammeln und Speichern von Wissen von vorrangiger Bedeutung, sondern die Identifikation von geeigneten Wissensträgern. Das System sollte den Aufbau und die Durchführung einer Kommunikationsbeziehung zu den Wissensträgern unterstützen. Für organisationsübergreifende Workflows bedeutet dies, dass man gezielt Wissensträger ansprechen können soll, von denen man weder Namen noch Abteilung weiß. Das System soll sowohl Prozesswissen als auch Funktionswissen managen. Als Prozesswissen bezeichnen wir das Wissen über den Prozessablauf, beteiligte Rollen, Personen, Organisationseinheiten, notwendige Daten und Ressourcen. Als Funktionswissen bezeichnen wir Wissen, welches für die Durchführung einzelner Prozessschritte notwendig ist. WfMS unterstützen normalerweise nur das Prozess-

und vernachlässigen das Funktionswissen. In einem B2B-Kontext verstehen wir unter Prozesswissensmanagement auch, dass der verteilte Wf-Entstehungsprozess unterstützt wird. Funktionswissen soll explizit gemacht werden können, um es entweder beim Scheduling oder zum Zwecke des Skillmanagement zu nutzen. Durch die erweiterte Funktionalität gegenüber traditionellen WfMS gibt es weitere Auswertungs- und Darstellungsmöglichkeiten. Hier ist eine offene Schnittstelle gefordert, so das beispielsweise Skillmanagement eingefügt werden kann.

### 3 Peer-to-Peer-Technologie

Die Peer-To-Peer-Technologie ist ein Spezialfall der traditionellen Client/Server Technologie, wobei im Idealfall jeder Knoten im Netz sowohl Client als auch Serverfunktionalität hat. Im Unterschied zum üblichen Client/Server-Ansatz sind somit alle Knoten prinzipiell gleichberechtigt. Um eine besondere Robustheit zu erreichen, agieren die Knoten (Peers) weitgehend autonom, d.h. sie können selbst entscheiden, sich z.B. vom Netz abzutrennen oder welche Dienste sie zur Verfügung stellen. Bei Kommunikation zwischen Peers gibt es keine zentrale Instanz, die die Nachrichten filtert und verteilt etc. Stattdessen kommunizieren die Peers „direkt“, was nicht ausschließt, dass andere Peers vermittelnd tätig sind, d.h. die Nachricht annehmen und weiterleiten. Entscheidend ist, dass es keine *zentrale* Vermittlungsstelle gibt. Typische Anwendungen im P2P-Bereich sind: File-Sharing, Instant-Messaging (IM), und kollaborative Anwendungen. Beim P2P-Filesharing werden die Daten direkt zwischen den Peers ausgetauscht und nicht in einem zentralen Server zwischengehalten. IM ermöglicht eine „getippte“ Kommunikation in „Echtzeit“, beim P2P-Messaging werden die Nachrichten wiederum direkt zwischen den Peers ausgetauscht. File-Sharing und IM sind beides Funktionen, die sinnvoll in einem kollaborativen Kontext eingesetzt werden. Andere Funktionen, die mit P2P-Technologie realisiert werden können, sind Chatrooms, elektronische Foren, Co-Authoring etc. Als spezifische Ausprägung des P2P-Ansatzes haben wir die JAVA-Implementierung des JXTA-Protokolls gewählt, welche im Rahmen eines von SUN initiierten Projekts entstanden ist. JXTA erlaubt es unter anderem, an einem Peer Dienste (Services) anzubieten und diese durch sog. Advertisements bekannt zu geben. Andere Peers können diese Dienste suchen und mit dem Diensterbringer über eine virtuelle Verbindung (Pipe) kommunizieren. Die Verbindungen werden virtuell genannt, weil sie auf konkrete Protokolle wie TCP oder HTTP abgebildet werden. Pipes werden mit Endpunkten verbunden, die wiederum von konkreten IP-Adressen abstrahieren. Auf diese Weise wird ein unabhängiger Adressierungsschema etabliert, welches sicherstellt, dass die Kommunikation auch dann reibungslos funktioniert, wenn die IP-Adressen dynamisch zugeteilt werden. Gateway-Peers erlauben die Überwindung von Firewalls, indem sie TCP- mit HTTP-Pipes verbinden. Firewalls sind in den meisten Fällen so konfiguriert, das ausgehende HTTP-Anfragen wie sie von üblichen Webbrowsern erzeugt werden, zugelassen sind. Eingebunden in die JAVA-Implementierung von JXTA ist eine Security-Suite, die Verschlüsselung, Peer-Authentifizierung etc anbietet.

## 4 Architektur des neofonie B2B-WfMS :flow

Die grobe Architektur der Wf-Komponente „neofonie architect:flow“ besteht aus zwei P2P-Ebenen. Organisationen, die an Wfs partizipieren, werden durch Peers (Typ A) repräsentiert, die bei der Bearbeitung von Wfs direkt miteinander kommunizieren. Jeder dieser Peers ist auf tieferer Ebene ein Typ-B-Peer in dem unternehmensinternen P2P-Netz. Unternehmensinterne Peers schließlich bieten Dienste an, die bei der Bearbeitung eines Wfs in Anspruch genommen werden. Siehe dazu Abbildung 1 linke Seite.

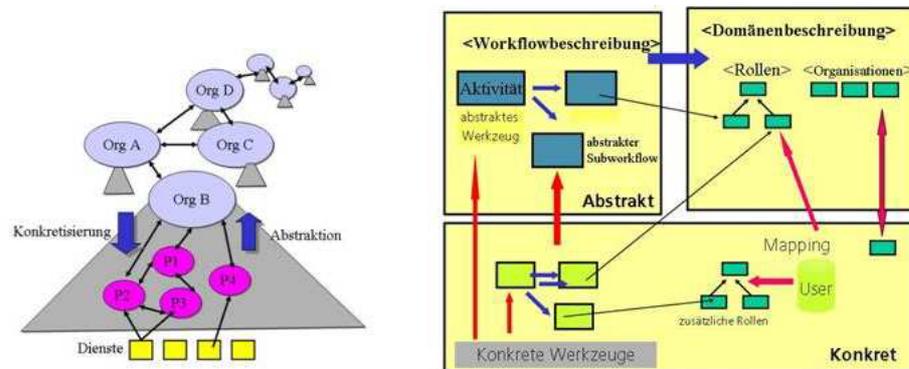


Abbildung 1.

Das System ist XML-basiert, d.h. sowohl das Wf-Schema, als auch die Wf-Instanz wird als XML-Dokument repräsentiert. Jede Wf-Instanz enthält die zu bearbeitenden Objekte als Payload (virtuelle Umlaufmappe) und wandert im Zuge der Bearbeitung von einem Peer zum nächsten. Dabei kann die Instanz bei einem Split auf verschiedenen Peers verteilt und bei synchronisierenden Joins wieder an einem Peer zusammengeführt werden. Die Auswahl der Peers wird entsprechend des Wf-Schemas von der Wf-Engine vorgenommen. Dabei werden gegebenenfalls die Ergebnisse der bereits vollzogenen Aktivitäten berücksichtigt, z.B. bei Fallunterscheidungen. Die Menge der geeigneten Peers wird durch Ausnutzung des JXTA-Protokolls ermittelt: Aktivitäten entsprechen auf JXTA-Ebene Diensten und werden daher per Advertisement bekanntgegeben und können gesucht werden. Entscheidend zur Umsetzung vieler der genannten Anforderungen ist das zugrundeliegende *Domänenkonzept*, welches sehr ähnlich zu dem in [7] beschriebenen Konzept ausgelegt ist. Jeder Wf wird einer Domäne zugeordnet; innerhalb einer Domäne können verschiedene Wfs definiert sein. In der Domänenbeschreibung werden die partizipierenden Organisationen genannt und die vorkommenden Rollen deklariert. In dem Wf-Schema wird die Domäne benannt und für jede Aktivität angegeben, unter welcher Rolle sie ausgeführt

wird. Weiterhin wird dort für jede Aktivität ihre Signatur als XML-Schema beschrieben. Damit werden die Ein- und Ausgabeparameter festgelegt, sowie der Teil des Payloads identifiziert, auf den zugegriffen wird. Ergänzend wird die genaue Zugriffsart (z.B. erzeugend, nur lesend) pro Aktivität festgelegt. Die feingranulare Zugriffskontrolle stellt sicher, dass ein Benutzer nur dann die spezifizierten Zugriffsrechte erhält, wenn er in der Rolle aktiv ist und die Aktivität auch gerade ausführt. (vergl. [7]) Der Standard XMLDSIG wird verwendet, um die jeweils bearbeitenden Teile der Wf-Instanz userspezifisch zu signieren. Einige der Sicherheitsanforderungen können mithilfe der JXTA-Security-Suite erfüllt werden, so ist beispielsweise TLS als Transportverschlüsselung eingebunden, die Authentifikation von Peers wird durch Peer-Zertifikate sichergestellt. Eine ausführlichere Darstellung zu den Sicherheitsanforderungen und ihre Realisierung findet sich in [8]. Transaktionale Wf werden durch Einbindung des Business Transaction Protocols unterstützt. Um die Autonomieanforderungen zu erfüllen sind Wf-Schemas in mehrfacher Weise unterspezifiziert. Beim Übergang von Typ-A auf Typ-B-Peers, also beim Eintritt in ein Unternehmen wird bezüglich der unterspezifizierten Anteile eine Konkretisierung vorgenommen. Beim Austritt aus dem Unternehmen wird umgekehrt durch einen Abstraktionsschritt die Konkretisierung wieder rückgängig gemacht. Der Zusammenhang ist in der Abbildung 1 auf der rechten Seite dargestellt. Im Einzelnen betrifft das folgende Bestandteile:

- **Rollen abstrahieren von konkreten Benutzern und Organisationsstruktur.** In dem Wf-Schema wird für jede Aktivität die Rolle angegeben, unter der sie ausgeführt wird. Rollen werden in der Domänenbeschreibung abstrakt definiert. Die Rollenbeschreibung besteht dort nur aus einem Namen, einer natürlichsprachlichen Beschreibung sowie der Angabe von Superrollen. Dadurch, dass die Rollen domänenspezifisch definiert sind, haben sie keinen Bezug zur Organisationsstruktur der beteiligten Organisationen. Die Anbindung von Usern geschieht unternehmensintern durch eine Zuordnung von Usern zu Rollen. An dieser Stelle kann die Benutzerverwaltung durch LDAP, NT oder UNIX-Systeme eingebunden werden. Diese Systeme fassen Benutzer zu Gruppen zusammen. Häufig reflektieren die Gruppen auch die Organisationsstruktur. Durch Ausnutzung des Gruppenkonzepts können bei der Zuordnung nicht nur einzelne Benutzer, sondern Gruppen, und damit gegebenenfalls organisatorische Einheiten, berücksichtigt werden.
- **abstrakte Werkzeuge.** Ähnlich wie MIME-Typen erlaubt das System das verwendete Werkzeug im Schema zunächst abstrakt festzulegen, z.B. „Textverarbeitung“. Die Konkretisierung (z.B. „MS-Word“) findet in diesem Fall für die einzelnen Peers statt. Aus Sicherheitsgründen können für bestimmte Aktivitäten auch spezifische Programme fest vorgeschrieben werden, die nicht mehr verändert werden können.
- **unterspezifizierte Sub-Workflows.** Unternehmensinterne Teilworkflows können in dem Wf-Schema unterspezifiziert bleiben, die konkrete Realisierung wird in einem extra Dokument modelliert und ist nach außen nicht sichtbar. Auf diese Weise können verschiedene Organisationen mit unter-

schiedlichen Realisierungen arbeiten, ohne das Einzelheiten der Realisierung bekannt gegeben werden müssen.

## 5 Wissensmanagementunterstützung

In diesem Abschnitt soll auf spezielle Systemfunktionen eingegangen werden, die :flow besonderes für Wissensmanagement geeignet machen, insbesondere im Kontext von den Anforderungen bei interorganisationalen Workflows.

### 5.1 Management von Prozesswissen

Jeder Wf ist einer Domäne zugeordnet. Für jede Domäne gibt es eine Metadomäne und einen Meta-Wf, die die kollaborative Entwicklung eines Wfs ermöglichen. Der Payload des Meta-Wfs besteht aus dem sich in der Entwicklung befindlichen Wf-Schema des Objekt-Wfs. Dadurch, das der Entstehungsprozess selbst als Wf beschrieben ist, ist er äußerst flexibel und kann leicht an die jeweiligen spezifischen Anforderungen angepasst werden. Dieser Ansatz ist wiederum für die geografisch verteilte Entwicklung sinnvoll, wie sie bei B2B-Workflows auftritt. Weiterhin werden Wf-Schemas in einem Repository abgespeichert, auf dem fallbasierter oder kategorienbasierter Zugriff besteht. Damit ist bei der Entwicklung von neuen Wfs das gesamte explizite Prozesswissen verfügbar (Kodifizierungsstrategie). Zusätzlich können die jeweils Verantwortlichen kontaktiert werden (Personifizierungsstrategie).

### 5.2 Management von Funktionswissen

Ähnlich wie in [9] dargestellt, erlauben wir für jede Rolle/Aktivität/Anwendung die explizite Annotierung von benötigten Fähigkeiten/Fertigkeiten/Kenntnissen (Skills). Zwischen Anwendung, Aktivität, Rolle besteht dabei eine Vererbungsrelation. Diese Modellierung kann benutzt werden um Wissenslandkarten über Wfs-Rolle-Skills zu generieren oder einen Erfahrungsindex der Mitarbeiter aufzubauen oder andersrum bei der Auswahl des Sachbearbeiters vom System berücksichtigt werden. Der GP liefert den Kontext für Wissensprozesse. Dieses kann im System ausgenutzt werden, wenn man die Wissensprozesse explizit mitmodelliert, und den Kontext zugänglich macht. Z.B. können automatisch Suchmasken entsprechend der aktuellen Instanz ausgefüllt werden. Ein ähnliches Vorgehen ist in [6] dargestellt. Ein Wf-Schema hat möglicherweise viele Instanzen. Diese werden in einer von neofonie entwickelten XML-Datenbank abgelegt. Fallbasiertes Retrieval liefert zu der aktuellen Instanz die ähnlichsten Fälle, die in der Vergangenheit bearbeitet wurden.

### 5.3 Kommunikationsmanagement

Das System unterstützt eine Vielzahl von Kommunikationsmöglichkeiten. Der Payload kann zur Laufzeit um Kommentare, Fragen und Anregungen erweitert

werden, die dann weiteren Bearbeitern zugänglich sind. Überhaupt es möglich derartige Fragen, Anmerkungen zu formulieren und über unterschiedliche Kanäle zu verteilen. Die P2P-Technologie macht es leicht, Anwendungen wie IM, Gruppen-Chat und Diskussionsforen zu realisieren. Natürlich ist auch eMail verfügbar. Durch das (Meta)Domänen- und Rollenkonzept sind vielfältige Adressatenkreise verfügbar. **Instanzspezifisch:** Es können gezielt Bearbeiter kontaktiert werden, die einen bestimmten Arbeitsschritt in der vorliegenden Instanz bearbeitet haben. Es kann eine Nachricht an die Gruppe aller, die bisher die Instanz bearbeitet hat, gesendet werden. Schließlich kann eine Nachricht an alle geschickt werden, die diese Instanz bereits bearbeitet haben oder zukünftig bearbeiten werden. Man beachte, dass sich die Adressierung nicht nur auf die aktuelle Instanz bezieht, sondern für beliebige Instanzen gilt. Wenn das oben genannte fallbasierte Retrieval beispielsweise eine ähnliche Instanz geliefert hat, kann der Sachbearbeiter den Kollegen kontaktieren, der den entsprechenden Arbeitsschritt bei dem ähnlichen Fall bearbeitet hat. **Schemaspezifisch:** Es kann eine Nachricht an alle geschickt werden, die potentiell Instanzen des Schemas bearbeiten. Der Empfängerkreis kann durch Auswahl von bestimmten Rollen eingeschränkt werden. Auf diese Weise kann z.B. eine Diskussion über Probleme bei der Bearbeitung einer bestimmten Aktivität geführt werden, wobei alle Personen adressiert werden, die auf Grund ihrer Rollenzuordnung die Aktivität durchführen können. **Meta-Wf-Spezifisch:** Es kann eine Nachricht an alle geschickt werden, die das Wf-Schema entwickelt haben. Hier können Fragen oder Verbesserungsvorschläge für den Workflow diskutiert werden. **Domänenspezifisch:** Es kann eine Nachricht an alle gesendet werden, die eine bestimmte Rolle innerhalb der Domäne innehaben. Das ist eine workflowübergreifende Kommunikation, da sich innerhalb einer Domäne mehrere Workflows befinden können. **Metadomänenspezifisch:** Es kann eine Nachricht an die Metadomäne geschickt werden, d.h. an diejenigen, die die vorkommenden Rollen festgelegt haben.

Für alle oben genannten Adressierungsvarianten kann durch Festlegung auf bestimmte Organisationen der Empfängerkreis weiter eingeschränkt werden. Damit ist es beispielsweise möglich eine Nachricht an die unternehmensinternen Bearbeiter eines Workflowschemas zu schicken. Abschließend wollen wir die Kommunikationsmöglichkeiten an Hand eines Szenarios erläutern: Ein international tätiges Kreditunternehmen verfasst in der Zentrale Kreditrichtlinien, die dann zu Umsetzung an die einzelnen Länderniederlassungen gehen. Die Kreditrichtlinien lassen noch einen gewissen Spielraum bei der Umsetzung zu. Der italienische Sachbearbeiter einer bestimmten Kreditrichtlinie kann jetzt seinen Kollegen in Deutschland kontaktieren, der genau dieselbe Richtlinie umsetzen muss, ohne den Namen des Sachbearbeiters, die Abteilung etc wissen zu müssen. Die Flexibilität der Kommunikation kann nochmals gesteigert werden, wenn die Kommunikationsbeziehungen wiederum als Wf modelliert werden, damit ist es z.B. möglich Folgendes auszudrücken: Sende eine Frage an eine bestimmte Gruppe, wenn drei Antworten eingegangen oder Zeitschranke überschritten ist, dann lösche die Frage aus der Liste der Empfänger, die die Nachricht noch nicht gelesen haben.

## 6 Bewertung der P2P-Technologie

Die Bewertung des Einsatzes von P2P-Technologie ist immer ein kritischer Punkt. Natürlich kann man ein System mit gleicher Funktionalität auch auf eher traditionellem Wege (Client-Server-Architektur) realisieren. In der beschriebenen Architektur kommt das Dokument zu dem Benutzer, der es zu bearbeiten hat, in einer Client-Server Architektur ist es normalerweise andersherum: der Nutzer wendet sich mit Hilfe eines Clientprogramms an den Server und geht damit zum Dokument. Das in dem Zusammenhang der Bewertung von P2P-Systemen häufig vorgetragene Argument der strukturellen Analogie ist aus unserer Sicht bestenfalls motivierend für den Einsatz von P2P. Eine Bewertung müsste entlang vorher definierter Kriterien vollzogen werden, und zwar vergleichend zwischen Systemen mit verschiedenen Architekturen. Abgesehen von dem großen Aufwand der mehrfachen Realisierung ist ein solcher Vergleich immer noch nur begrenzt aussagefähig; Schließlich werden konkrete Systeme auf Basis unterschiedlicher Architekturmodelle verglichen und nicht die Architekturmodelle an sich. Letztendlich wird sich eine Art empirische Bewertung ergeben, die sich auf Grund von Erfahrungen beim Design von Systemen entwickelt. Eigentlich sehen wir in P2P auch nur eine Spielart des Client-Server-Ansatzes und Systeme werden in vielen Fällen hybrid sein, also Aspekte haben, die man eher einer Client-Server-Architektur zuordnen würde. Das ist im vorliegenden Fall nicht anders. Einige Aufgaben bei einer WfMS erfordern eher eine zentrale Komponente, und widersprechen damit dem P2P-Gedanken. Dazu gehören eine Überwachungskomponente, die es erlaubt zu jeder Wf-Instanz Informationen zur Verfügung zu stellen, welche Aktivitäten gerade ausgeführt werden. Andere Beispiele sind die Repository-Komponente sowie Unterstützung von Transaktionen. In unserem System unterscheiden wir deshalb als Hostrechner für Peers Server von Arbeitsplatzstationen von Mitarbeitern. Einige Dienste werden auf „immer verfügbaren“ Servern lokalisiert. Es gibt einige Bereiche, wo sich P2P in der Ausprägung von JXTA als günstig erweist. Als Vorteil kann verbucht werden, dass P2P-Architekturen dem Autonomiebedürfnis der Organisationen stärker gerecht werden: Es gibt keinen zentralen Server außerhalb der Organisation, von dessen Ausfallsicherheit die Organisation abhängig ist. Ist auf der anderen Seite gewünscht, die Administration des WfMS outzusourcen, ist möglicherweise aber doch ein zentraler Server erwünscht. Ein Vorteil der verwendeten P2P-Technologie ist die Unterstützung von mobilen Anwendern und Geräten. Das betrifft den Aspekt der Offlinebearbeitung sowie der Kommunikation. Ein User kann sein Notebook unter beliebiger IP-Adresse mit dem Netz verbinden und einen Task annehmen. Die Bearbeitung des Tasks kann offline geschehen, sofern das benötigte Werkzeug lokal vorhanden ist. Sobald das Gerät wieder IP-Kontakt hat, wird das Ergebnis übermittelt. JXTA überlagert bestehende Netzwerkarchitekturen mit einem virtuellen Overlaynetzwerk. Auf dieses Weise ist es prinzipiell möglich auf Protokolle für Ad-Hoc-Vernetzung wie Bluetooth aufzusetzen. Weiterhin erscheint P2P-Technologie gut geeignet für die Realisierung von ergänzender kollaborativer Kommunikationsunterstützung, wie sie aus Wissensmanagementsicht erwünscht

ist. Die Beurteilung der Fähigkeit von JXTA, durch Firewalls durch Ausnutzung von http zu tunneln, wird je nach Security-Policy unterschiedlich ausfallen.

## 7 Ähnliche Ansätze und Ausblick

Wir haben ein B2B-WfMS vorgestellt, welches auf Basis von P2P-Technologie implementiert ist und besondere Unterstützung im Bereich WM bietet. Es gibt eine Reihe von Arbeiten im Bereich prozessorientiertes WM [2]. Für System :flow kennzeichnend ist die Unterstützung in der Designphase sowie das umfangreiche Kommunikationsmanagement. Informationen über P2P-basierte WfMS findet man bisher nur wenige, z.B. in [10]. Unter der URL [s2s.neofonie.de](http://s2s.neofonie.de) [12] findet man unter dem Stichwort „S2S Interaktiv“ eine Beispielanwendung von :flow, bei der eine Kommunikationsbeziehung explizit modelliert ist. Eine naheliegende Weiterentwicklung des Systems liegt darin, neben dem Workflow auch die Softwarewerkzeuge über das JXTA-Netzwerk zu verschicken.

## Literatur

1. Aalst, W.M.P., Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: :Workflow Pattern (2002)
2. Abdecker, Andreas, Hinkelmann, Knut, Maus, Heiko, Müller, Hans Jürgen *Geschäftsprozessorientiertes Wissensmanagement* Springer Verlag, ISBN 3-540-42970-0, (2002)
3. Böhm, Markus: *Entwicklung von Workflow-Typen* Springer Verlag, ISBN 3-540-66394-0 (2000)
4. Bussler, Christoph: *B2B Integration. Concepts and Architecture*. Springer Verlag, ISBN 3-540-43487-9 (2003)
5. Hansen, M.T., Nohria, N., Tierney, T.: *Whats's your Stragegy for Knowledge Management* In: Harward Business Review, March-April 1999, pp.106-116
6. Hinkelmann, Knut, Kraragiannis Dimitris, Telesko Rainer: *PROMOTE-Methodologie und Werkzeuge für geschäftsproessorientiertes Wissensmanagement* in [2]
7. Kang, Myong ,H., Park, Joon s., Froscher, Judith N.: *Access Control Mechanism for Inter-organisation Workflow*
8. Küssner, U.: *Sicherheitsanforderungen bei B2B-Workflows: Realisierung mit JXTA-Security und XML-Security-Standards*. JAVA-Spektrum Ausgabe 1, Januar/Februar '04 (2004)
9. Nägele, Rainer, Schreiner, Peter: *Potenziale und Grenzen von Business Process Management Tools für geschäftsprozessorientiertes Wissensmanagement* in [2]
10. Noll, J.: *Process Enactment in Virtual Software Organisations*
11. Nonaka, Ikujiro, Takeuchi, Hirotaka; *Die Organsisation des Wissens* Campus Verlag, ISBN 3-593-53643-0 (1997)
12. Wertlen, Ron: DFN Science-to-Science: Peer-to-Peer Scientific Research Terena Networking Conference 2003 (2003), Zagreb, Coratia

# Integration heterogener Applikationstypen durch Workflow-Management-Technologie

Thomas Bauer

DaimlerChrysler Research and Technology, Abt. RIC/ED, Postfach 2360, D-89013 Ulm  
Thomas.TB.Bauer@DaimlerChrysler.com

**Zusammenfassung.** In diesem Betrag wird ein EAI-Projekt mit einem etwas außergewöhnlichen Fokus vorgestellt: die Integration von Anwendungen von völlig unterschiedlichem Typ. Deren Kopplung ist erforderlich, da sie in demselben Anwendungskontext verwendet werden sollen. EAI-Technologie wird z.Zt. meist zur Daten- oder Funktionsintegration von solchen Anwendungen verwendet, die ähnliche Aufgaben erfüllen. Da die zu integrierenden Anwendungen kaum Gemeinsamkeiten aufweisen, ist eine Integration auf der Daten- oder Funktionsebene aber nicht möglich. Um im betrachteten Projekt dennoch eine Kooperation zu ermöglichen, wurde diese prozessbasierend umgesetzt. Unterstützt wird sie noch durch eine graphische Visualisierung von aktuell ausgeführten Prozessinstanzen.

## 1 Einleitung

Das im Folgenden betrachtete Anwendungsbeispiel wurde von der DaimlerChrysler-Forschung gemeinsam mit Geschäftsbereichen umgesetzt. Es wurde in der Fahrzeugentwicklung durchgeführt, der vorgestellte Ansatz lässt sich aber auf alle Anwendungsdomänen übertragen, in denen unterschiedliche Typen von Applikationen eingesetzt werden. Im vorgestellten Projekt muss das Projektmanagementsystem RPlan und ein eigenentwickeltes Controlling-Tool mit mehreren operativen Entwicklungssystemen aus dem Bereich Elektrik/Elektronik (E/E) integriert werden. Die ersten beiden dienen der Steuerung eines Projekts, wohingegen die Operativsysteme den aktuellen Bearbeitungszustand dieses Projektes enthalten.

Ziel der Systemintegration ist, den aktuellen Bearbeitungszustand dem Projektmanagement zugänglich zu machen. Dadurch kann er Abweichungen eher erkennen und den Projektplan früher anpassen. Außerdem sollen die Entwickler, die mit den operativen Systemen arbeiten, mit Planungsdaten (Projektabläufe und -termine) versorgt werden. Dies erlaubt den Sachbearbeitern, ihre Aufgaben besser in das Gesamtprojekt einzuordnen und ihre eigentlichen Aufgaben vorausschauender zu planen. Um unnötigen Zusatzaufwand zu vermeiden, ist eine Nebenbedingung, dass mehrfache Eingaben derselben Daten vermieden werden, und Daten der verschiedenen beteiligten Systeme automatisch abgeglichen (d.h. ausgetauscht) werden. Schließlich existiert noch die Anforderung, dass das resultierende System sehr leicht zu bedienen ist und die Zusammenhänge einfach (wenn möglich graphisch) dargestellt werden.

Als Hauptproblem stellte sich dabei heraus, dass existierende EAI-Ansätze eher auf Applikationen mit ähnlichem Verwendungszweck (z.B. Produktdatenmanagement- und Stücklistenverwaltungssysteme) abzielen. Eine Integration völlig unterschiedlicher Anwendungstypen ist mit diesen Ansätzen hingegen nicht möglich, da weder bei den Daten noch bei den Funktionen eine nennenswerte Überlappung existiert: Projektmanagementsysteme und E/E-Entwicklungswerkzeuge haben keine gemeinsame Funktionalität. Die einzigen Daten, welche in beiden Systemen vorhanden sind, sind die tatsächlichen Bearbeitungszeiten und -zustände. Doch selbst diese liegen nicht im Fokus der Entwicklungssysteme, sondern finden sich eher implizit in den Historieninformationen. Da diese Daten nur einen extrem geringen Bruchteil der von diesen Systemen verwalteten Informationen darstellen, ist auch eine klassische Datenintegration hier nicht sinnvoll.

Deswegen wurde in dem vorgestellten Projekt die Integration mittels eines Prozesses realisiert. Dieser ermöglicht den Datenaustausch zwischen den heterogenen Applikationen. Durch eine Prozessvisualisierung wird den Beteiligten zusätzliche Information über Projektstruktur und -status bereitgestellt, sowie detaillierte Daten zu dem Bearbeitungszustand der einzelnen Projektaufgaben. Hierfür und zur Unterstützung der Prozessbearbeitung werden Daten aus den verschiedenen integrierten Anwendungssystemen verwendet. Ausgelesene sowie bei der Prozessausführung erzeugte Daten werden des Weiteren auch in die Applikationssysteme geschrieben, so dass ein Datenabgleich realisiert wird. Die Systeme selbst bleiben aber unangetastet.

## 2 Verwandte Arbeiten

Klassische Ansätze für EAI [10, 11, 14] versuchen Applikationen auf Daten- oder Funktionsebene zu integrieren. Wie schon erwähnt ist ein solcher Ansatz bei der gegebenen Aufgabenstellung nicht anwendbar, weil die betroffenen Systeme zu wenig Überlappung bezüglich Daten und Funktionalität haben.

Die Verwendung von Portalen [13] ist ebenfalls nicht zielführend, da sich durch diese lediglich ein gemeinsamer Einstiegspunkt für mehrere Anwendungssysteme realisieren lässt, und häufig auch nur lesender Zugriff unterstützt wird. Ein Datenaustausch zwischen den Systemen kann durch diese Vorgehensweise nicht realisiert werden.

Ein Data-Warehouse [6] kann den Benutzern Informationen zur Verfügung stellen, die aus mehreren Quellsystemen stammen. Allerdings wird der direkte Datenaustausch zwischen den Quellsystemen durch diese Technologie nicht unterstützt. Deshalb ist es z.B. nicht möglich, den tatsächlichen Bearbeitungszustand in ein Projektmanagementsystem zurückzuschreiben, so dass er für die Projektplanung unmittelbar zur Verfügung steht. Auch eine prozessorientierte (graphische) Sicht auf den Entwicklungsablauf wird durch eine solche Datenzusammenführung nicht unterstützt, so dass dieser Ansatz hier nicht trägt.

Workflow-Management-Systeme (WfMS) [7, 9] wurden eigentlich nicht zur Realisierung von EAI-Lösungen, sondern zur Steuerung von Geschäftsprozessen entwickelt. Da in diese Prozesse auch Geschäftsanwendungen eingebunden werden müssen, ermöglichen WfMS normalerweise den Aufruf von Applikationen als Teilschrittprogramme. Häufig werden sogar schon vorgefertigte Adapter zu Standardap-

pplikationen wie SAP oder CICS angeboten (z.B. in WebSphere MQ Workflow [8]). Dadurch bieten WfMS eine gute Plattform zur Integration von heterogenen Applikationen, falls die Integration durch einen „People-Workflow“ erfolgen soll. Da WfMS zukünftig hoffentlich zunehmend anspruchsvolle Funktionalität wie Verteilung und Dynamik [2, 3, 4] unterstützen werden, stellen sie eine strategisch günstigere Implementierungsplattform als z.B. Lotus Notes oder WebSphere MQ dar. Außerdem helfen sie schon heute den Implementierungsaufwand zu reduzieren, da WfMS Funktionalität wie Prozesssteuerung, Arbeitslistenverwaltung oder Monitoring bereits realisieren.

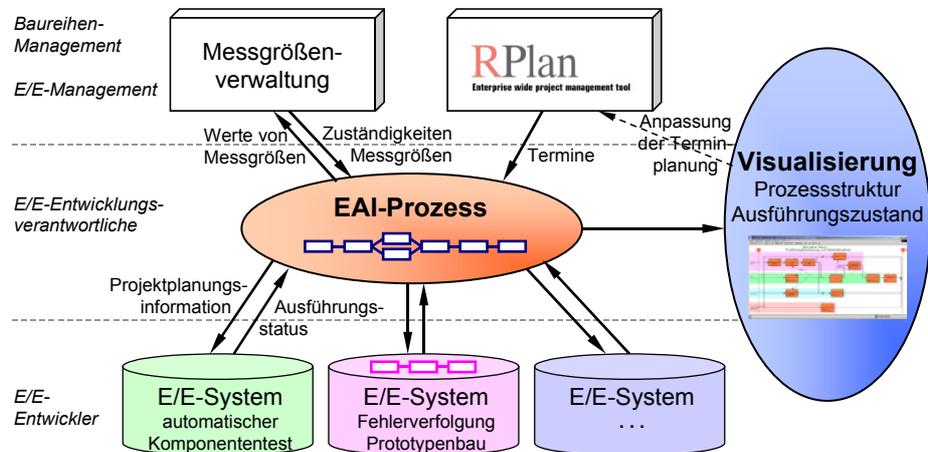
Auch die mit EAI-Technologie zu integrierenden Applikationen können Prozesslogik enthalten. Außerdem ermöglichen EAI-Tools häufig die Definition von „Microflows“ zur Steuerung des Zugriffs auf Funktionen verschiedener Anwendungssysteme bei der Ausführung einer Geschäftsfunktion. Diese werden allerdings automatisch, d.h. ohne Beteiligung von Personen, ausgeführt. In beiden Fällen befindet sich der Prozess aber eher auf einer semantisch niedrigeren Ebene und ist deshalb im Gegensatz zu WfMS nicht für eine stark prozessorientierte Applikationsintegration geeignet.

### 3 Prozessbasierende Applikationsintegration

Ziel des Projektes war es, die folgenden Applikationssysteme (siehe Abb. 1) zu integrieren:

- Das Projektmanagementsystem RPlan von Actano [1] verwaltet die Abläufe, d.h. die Projektaufgaben und deren Abhängigkeiten, des Entwicklungsprojekts sowie die zugehörigen Termine auf einer sehr detaillierten Ebene.
- Das Messgrößenverwaltungssystem dient zum Controlling des Gesamtprojektes auf hohem Abstraktionsniveau. Es verwaltet Quality Gates, für deren Durchschreiten definierte Messgrößen bestimmte Bedingungen erfüllen müssen. Die Festlegung der entsprechenden Messwerte erfolgt ebenso mit diesem System, wie die Verwaltung der Verantwortlichkeiten für Messgrößen und -werte.
- Im Elektrik/Elektronik (E/E) Bereich werden unterschiedliche Systeme zur Unterstützung der Produktentwicklung eingesetzt. Aufgrund ihrer völlig unterschiedlicher Aufgaben gibt es kaum Beziehungen zwischen ihnen, die gespeicherten Daten lassen aber jeweils Rückschlüsse auf den aktuellen Stand der Produktentwicklung zu. Hier seien exemplarisch zwei von ihnen vorgestellt:
  - Das automatische Testsystem für E/E-Aufbauten verwaltet u.a. die erzeugten Testergebnisse, welche wiederum Rückschlüsse auf den Reifegrad von E/E-Komponenten (z.B. Steuergeräte) zulassen.
  - In der Fehlerverfolgung des Prototypenbaus wird im Fall von entdeckten Fehlfunktionen ein Workflow ausgelöst, welcher der Beseitigung und Dokumentation des Problems dient. Die Anzahl und der Zustand dieser Prozesse erlauben einen Rückschluss auf den Reifegrad des E/E-Gesamtsystems.

Es ist offensichtlich, dass eine direkte Daten- bzw. Funktionsintegration allenfalls für jeweils sehr wenige dieser Systeme und nur für kleine Teilfunktionalitäten möglich ist, da die Systeme zu extrem unterschiedlichen Aufgaben dienen. Eine solche Integ-



**Abb. 1.** Verwendung und Datenaustausch zwischen den integrierten Anwendungssystemen

ration anzustreben wäre nicht sinnvoll. Teilweise ist ein Datentransfer zwischen den Systemen auch überhaupt nicht automatisierbar: So kann aus dem zeitlichen Verlauf der Fehlerprozess-Auslösung im Prototypenbau unter Berücksichtigung der jeweils unterschiedlichen Schwere des Fehlers – die nur aus dessen textueller Beschreibung ersichtlich ist – zwar auf den Bearbeitungszustand der Aufgabe „Entwicklung von in Fahrzeug erprobungswürdigen E/E-Komponenten“ geschlossen werden, aber sicherlich nicht automatisch.

Da die semantisch tieferliegenden Ebenen ausscheiden, muss die Systemintegration über den Geschäftsprozess erfolgen. Dessen Ausführung erlaubt das automatische Sammeln von Daten ebenso wie eine menschliche Interaktion, dort wo sie notwendig ist (z.B. zur Aggregation von Daten). So kann automatisiert von mehreren Aktivitäten aus auf verschiedene E/E-Systeme zugegriffen werden. Ebenso können Projektmanagementinformationen aus RPlan gelesen werden und E/E-Applikationen gezielt zur Verfügung gestellt werden. In [5] haben wir bereits aufgezeigt, dass bei unterschiedlichem Abstraktionsniveau der Projektplanung und der operativen Tasks bzw. Workflows eine Zwischenschicht für die Informationszuordnung und Aggregation erforderlich ist. Es hat sich herausgestellt, dass diese im vorliegenden Szenario nur prozessorientiert realisiert werden kann. Um den Aufwand hierfür zu reduzieren und die Flexibilität zu erhöhen, wurde dies mit dem WfMS Lotus Workflow [12] durchgeführt.

Die Aktivitäten des Workflows werden i.d.R. von sog. Funktionsgruppensprechern bearbeitet, die jeweils für die Entwicklung von logisch zusammengehörigen E/E-Komponenten verantwortlich sind. Diese legen im Rahmen dieses Prozesses auch die Werte von Teilmessgrößen fest, aus denen sich dann die Messgrößenwerte ergeben. Die Messgrößen werden von der in Abb. 1 entsprechend benannten Applikation verwaltet. Einige Prozessschritte lesen von ihr die Beschreibungen und Zuständigkeiten von Messgrößen und schreiben nach der Bearbeitung der entsprechenden Aktivitäten auch Werte von Messgrößen zurück. Durch diese Vorgehensweise wird verhindert,

dass ein Benutzer mit mehreren Applikationen arbeiten muss. Die Oberfläche des WfMS und die entsprechenden Masken als einzige Benutzerschnittstelle für die Funktionsgruppensprecher tragen deutlich zur Akzeptanz des Ansatzes bei.

#### 4 Visualisierung von Prozessinstanzen

Das vorgestellte Projekt zeigt, wie eine Applikationsintegration mit Hilfe eines WfMS durchgeführt werden kann. Dies ist sicherlich nicht das erste Projekt, in dem diese Vorgehensweise gewählt wurde, da EAI-Systeme und WfMS deutliche Überlappungen ihrer Funktionalität aufweisen. Allerdings wurde im betrachteten Projekt auch noch ein Nebeneffekt genutzt, der sich als großer Vorteil herausstellte: Der bisher nur implizit gelebte Prozess wird nun explizit verfügbar, und das einschließlich seines aktuellen Abarbeitungszustandes. Dies bildet eine ideale Basis zur Information aller Prozessbeteiligten. So dient der Ausführungszustand natürlich der Projektleitung zur Planung und zum Controlling. Die Prozessvisualisierung wird aber auch zur Information der Sachbearbeiter verwendet: Die Entwickler kennen meist weder die Projektpläne und Termine eines Fahrzeugprojektes, noch den abstrakten Gesamtprozess. Deshalb ist es ihnen nicht möglich, ihre konkreten Aufgaben in den Gesamtkontext einzuordnen. Dies führt wiederum dazu, dass sie auf sich zukommende Verzögerungen nicht rechtzeitig erkennen und die Auswirkungen von selbst verursachten Verspätungen nicht einschätzen können.

Die Darstellung der im WfMS vorhandenen Information erfolgt idealerweise durch die graphische Visualisierung einer Prozessinstanz, da dies für „normale Benutzer“ wesentlich leichter verständlich ist als Listen, Kennzahlen oder Tabellen. Leider ist die Funktionalität der Visualisierungs- bzw. Monitoringkomponenten heutiger WfMS hierfür bei weitem nicht ausreichend. So zeigt Abb. 2 die Standard-Prozessvisualisierung von Lotus Workflow. Diese war den interviewten Endbenutzern deutlich zu unübersichtlich und ermöglicht zudem nicht, alle relevante Laufzeitinformation darzustellen. Diese Schwachpunkte sind aber keineswegs Lotus Workflow spezifisch; die entsprechende Funktionalität anderer WfMS ist sogar größtenteils noch geringer. So gelten eigentlich bei allen Standard-Visualisierungskomponenten die folgenden Beschränkungen:

- Es kann nur direkt diejenige Sicht auf den Workflow dargestellt werden, die bei der Modellierung erzeugt wurde. Dies ist aber eher ein technischer Blickwinkel, d.h. er beinhaltet auch automatische Aktivitäten (z.B. zur Zusammenführung von Dokumenten bei Joins) oder durch das Metamodell bedingte Subworkflows. Dadurch wird die Darstellung für nicht IT-geschulte Anwender zu unübersichtlich.
- Es können nur WfMS-interne Attribute wie Bearbeiter oder Startzeiten dargestellt werden. Es ist meist nicht möglich, Anwendungsdaten zu Visualisieren (z.B. Bezeichnungen von E/E-Komponenten, vom Bearbeiter angegebene Abarbeitungszustände laufender Aktivitäten oder voraussichtliche Beendigungstermine). Dies sind allerdings Daten, die für die Entwickler höchst relevant sind.
- Die graphische Darstellung ist häufig sehr unübersichtlich. Normalerweise kann diese auch nicht um Zusatzinformationen angereichert werden, welche die Übersichtlichkeit erhöhen würden. So könnte z.B. die Zusammengehörigkeit von Akti-

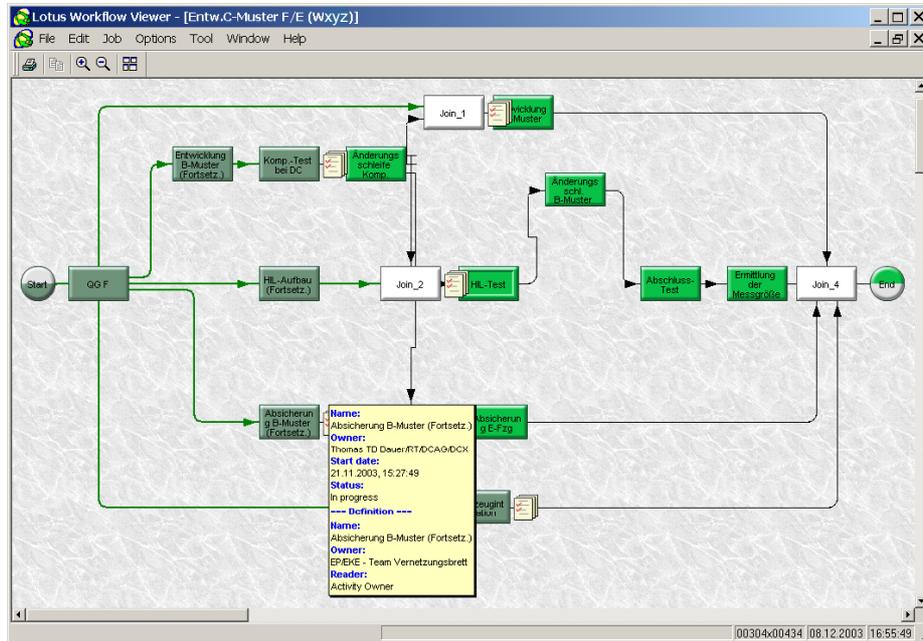


Abb. 2. Visualisierung einer Prozessinstanz in Lotus Workflow

vitäten eines Bereichs durch das Einblenden eines farbig ausgefüllten Rechtecks als Hintergrund deutlich gemacht werden. Auch Texte mit Zusatzinformationen und (gestrichelte) Pfeile zwischen Aktivitäten mit informellem Informationsaustausch wären geeignet, um die Übersichtlichkeit zu erhöhen.

Um eine derart erweiterte Darstellung zu ermöglichen, wurde im vorgestellten Projekt eine Visualisierungskomponente (siehe Abb. 3) selbst implementiert. Diese verwendet das bei der Workflow-Modellierung erzeugte (und dann exportierte) Prozessmodell, ein selbst definiertes Visualisierungsmodell und Anwendungsdaten, die teilweise außerhalb des WfMS gespeichert sind. Die Verbindung dieser Laufzeitdaten mit der Workflow-Instanz wird durch einen „Datenzugriffsabschnitt“ im Visualisierungsmodell definiert. Die gesamte Implementierung ist mit Ausnahme des Visualisierungsmodells anwendungsunabhängig. Dieses wird in Form einer XML-Datei festgelegt (deren DTD selbstverständlich ebenfalls anwendungsunabhängig ist), die eigentlich automatisch vom Modellierungswerkzeug des WfMS aufgrund von Darstellungsfestlegungen des Modellierers erzeugt werden sollte. Da heutige WfMS hierzu nicht in der Lage sind, wird die Generierung durch eine manuelle Erstellung simuliert. Dadurch entsteht natürlich das Problem, dass die Konsistenz des Visualisierungsmodells mit dem Workflow-Modell im Falle von dessen Änderung nicht garantiert werden kann.

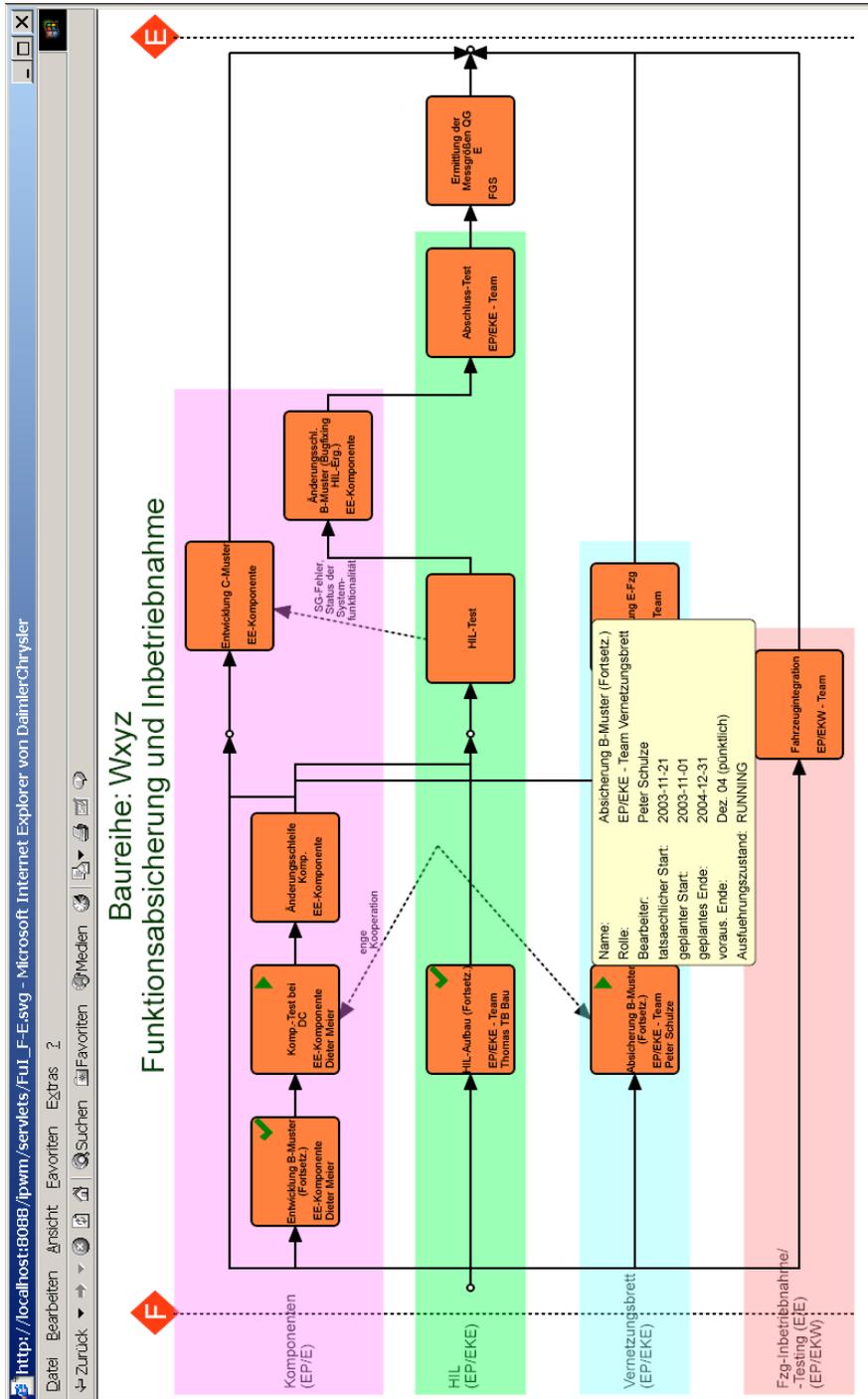


Abb. 3. Benutzergerechte Visualisierung einer Workflow-Instanz

Bei manchen Daten einiger Applikationssysteme erfolgt die Integration nicht durch ein direktes Schreiben einer automatischen Aktivität des Prozesses, sondern durch eine Benutzeraktion. Dies ist für eine Änderung der Pläne des Projektmanagementsystems auch der einzig gangbare Weg, weil eine automatische Umplanung aufgrund der Komplexität einer solchen Operation kaum vorstellbar ist und außerdem viel zu gefährlich wäre. Stattdessen wird die Visualisierung des aktuellen Ausführungszustandes (bei der auch aufgetretene Verzögerungen dargestellt werden) genutzt, um den Projektmanager bei dessen Analyse zu unterstützen. Dieser zieht dann selbst die entsprechenden Schlüsse und passt den Projektplan manuell an (angedeutet durch den gestrichelten Pfeil in Abb. 1).

## **5 Zusammenfassung und Ausblick**

Im vorgestellten Projekt war es die Aufgabe, Anwendungssysteme zu integrieren, für die dies wegen ihrer zu unterschiedlichen Einsatzgebiete mit den normalerweise verwendeten EAI-Methoden nicht möglich ist. Deshalb wurde eine Lösung gewählt, bei der die Integration über einen Geschäftsprozess erfolgt, der hierzu mittels eines WfMS implementiert wurde. Durch dessen Ausführung wird einerseits ein automatischer Datenaustausch zwischen den Applikationen realisiert, andererseits werden auch intellektuelle Entscheidungen der Prozessbeteiligten möglich. Um diese Entscheidungen auf solide Beine stellen zu können und zur Information aller Prozessbeteiligten, erwies sich die Visualisierung des ausgeführten Prozesses als sehr geeignet. Deshalb wurde eigens eine Visualisierungskomponente entwickelt. Für die Zukunft bleibt zu hoffen, dass die WfMS-Hersteller eine entsprechende Visualisierungsfunktionalität in ihre Systeme übernehmen werden, da ein sehr enger Bezug zur Modellierungskomponente des WfMS existiert. Damit wäre eine bedarfsgerechte Visualisierung ohne Zusatzaufwand für die jeweiligen Projekte verfügbar.

Abgesehen von dem Schwachpunkt der Visualisierung haben wir festgestellt, dass sich WfMS gut für die beschriebene Integration eignen, weil die Anforderungen an das Metamodell, die Workflow-Steuerung usw. im betrachteten Projekt nicht übermäßig hoch waren. Die Verwendung eines WfMS führt zu einem recht geringen Initialaufwand bei der Implementierung der prozessorientierten Applikation. WfMS bieten häufig schon vorbereitete Schnittstellen zu zahlreichen Typen von Anwendungssystemen. Abhängig von den konkret zu integrierenden Anwendungen können diese zu einer zusätzlichen Reduzierung des Umsetzungsaufwands führen. Da im konkreten Projekt hauptsächlich proprietäre Anwendungen ohne definierte Schnittstellen zu integrieren waren, blieb hier als einzige Lösung meist nur der Zugriff auf die Applikationsdatenbank zur Integration, so dass nicht von angebotenen Schnittstellen profitiert werden konnte. Dennoch hat sich das verwendete WfMS als sehr geeignete Implementierungsplattform erwiesen.

## Literatur

1. Actano: RPlan. <http://www.actano.de>
2. T. Bauer: Effiziente Realisierung unternehmensweiter Workflow-Management-Systeme. Dissertation Universität Ulm, Tenea-Verlag. (2001)
3. T. Bauer, M. Reichert, P. Dadam: Dynamische Ablaufänderungen in verteilten Workflow-Management-Systemen. Datenbank-Spektrum, Vol. 1(1). (2001) 68-77
4. T. Bauer, M. Reichert, P. Dadam: Intra-Subnet Load Balancing in Distributed Workflow Management Systems. Int. Journal of Cooperative Information Systems, Vol. 12(3). (2003) 295-323
5. T. Bauer, R. Siebert: Requirements and Methods for the Coupling of Project and Workflow Management Systems. In: Proc. ProSTEP iViP Conf. on Advances in Collaborative Product Creation, Dresden. (2003) 173-185
6. S.R. Gardner: Building the Data Warehouse. Communications of the ACM, Vol. 41(9). (1998) 52-60
7. D. Georgakopoulos, M. Hornick, A. Sheth: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases, Vol. 3(2). (1995) 119-152
8. IBM: MQSeries Workflow - Concepts and Architecture. (2001)
9. S. Jablonski, M. Böhm, W. Schulze: Workflow-Management: Entwicklung von Anwendungen und Systemen. dpunkt-Verlag. (1997)
10. M. Kaib: Enterprise Application Integration - Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Dissertation Universität Marburg, Deutscher Universitäts-Verlag. (2002)
11. W. Keller: Enterprise Application Integration - Erfahrungen aus der Praxis. dpunkt-Verlag. (2002)
12. Lotus: Domino Workflow Application Development. (2000)
13. J. Rütshlin: Ein Portal - Was ist das eigentlich? In: Proc. GI/OCG-Jahrestagung, Wien. (2001) 691-696
14. D. Serain: Middleware and Enterprise Application Integration. Springer. (2002)



# DÍGAME: A Vision of an Active Multidatabase with Push-based Schema and Data Propagation

Cristian Pérez de Laborda, Christopher Popfinger, and Stefan Conrad

Institute of Computer Science  
Heinrich-Heine-Universität Düsseldorf  
D-40225 Düsseldorf, Germany  
{perezdel, popfinger, conrad}@cs.uni-duesseldorf.de

**Abstract.** Sharing information in loosely coupled enterprises or virtual corporations demands a flexible and dynamic architecture, suitable for their individual data policies. The aim of this paper is to present the DÍGAME architecture, which balances both, local autonomy and a reasonable degree of information sharing. Therefore we combine the well known concept of loosely coupled multidatabases with more recent research in Peer-to-Peer or grid computing, satisfying the needs of modern intra- and inter-enterprise collaboration. In our architecture data and schema updates are propagated actively to subscribing component databases, without being managed by any central authority. This replication gives us the possibility to realize individual integration on each single peer.

## 1 Motivation

Since the first centralized databases found their way into the enterprises in the late 60s, the needs and requirements have changed towards a more distributed management of data. Today there are many corporations which possess a large amount of databases, often spread over different regions or countries and generally connected to a network. These local databases (components or component systems) typically raised in an autonomous and independent manner, fitting the special needs of the users at the local site. The design of the databases and the functionalities provided, intend to fulfil the aims of the departments. This leads to logical and physical differences in the databases concerning data formats, concurrency control, the data manipulation language or the data model [1]. An information system is required to integrate the information of these heterogeneous data sources to provide a global access.

One of the main challenges in the integration of data in such environments, is the autonomy of the participating data nodes (peers). This autonomy implies the ability to choose its own database design and operational behaviour [2]. Local autonomy is tightly attached to the data ownership, i.e. who is responsible for the correctness, availability and consistency of the shared data. Centralizing data means, to limit local autonomy and revoke the responsibility from the local administrator, which is not reasonable in many cases. The federated architecture

for decentralizing data has to balance both, the highest possible local autonomy and a reasonable degree of information sharing [3].

In this paper we introduce the vision of the DÍGAME architecture, a **D**ynamic **I**nformation **G**rid in an **A**ctive **M**ultidatabase **E**nvironment, which actively propagates data and schema updates over import/export-components between dynamically connectable data peers. This architecture offers a flexible and fail-safe information platform based on the data policies in organisations achieving a feasible trade-off between local autonomy and a reasonable degree of information sharing.

## 2 DÍGAME Architecture

### 2.1 Vision

The aim of this work is to introduce an architecture which allows the dynamic connection of data sources without restricting their local autonomy in order to share selected information. This union is based on Peer-to-Peer (P2P) concepts and operates without any central administrative instance.

The administrator of each peer makes a subset of its data accessible. Other peers are now able to integrate this data into their local databases, subscribing to a specific part of the data provided. Thereupon updates are propagated automatically to the subscribers by the data source, including both, data and schema modifications. Each data source of this dynamic information grid is herewith able to maintain an up-to-date replica of the required data and schema items. As there is no general rule for the integration of the replicated data, it has to be integrated individually by the administrator of each subscriber database.

Our architecture is especially designed to support dynamic intra- and inter-enterprise collaboration, by enabling each department involved to supply all relevant partners with the required information.

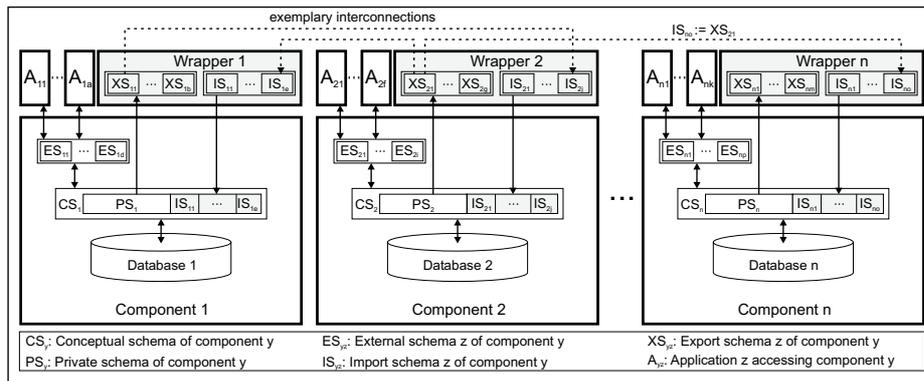
### 2.2 Components

We will now discuss the required components of the architecture using a case study, to draw up the benefits of our dynamic information grid. This example describes our approach to solve one of the multiple challenges concerning collaborative work: distributed information management.

Consider a worldwide operating company, planning the launch of a new product. To simplify our scenario, we assume that there are solely three departments involved in this business process, the executive board (management), the sales office and product engineering. Further departments may join this collaboration at any time. Each department manages its own database, to store the information for which it is responsible. The management produces basic data of the product. This includes deadlines, descriptions, workflows and additional objectives. This management information is substantial for the further product development and the work in the participating departments. The product engineering

uses a predefined part of that management data as basic conditions for the concrete implementation and technical realization. Local applications like CAD or measurement programs create additional data which has to be stored separately. According to the product engineering the sales department enriches the authoritative management data with concrete concepts for the oncoming product launch. Furthermore concrete development plans of the product engineering are required to prepare sales strategies. Both, sales and product engineering departments, concretize the strategic guidelines of the management in their specific assignment. To keep track of the costs and the progress of the project, it is indispensable for the management to access the product engineering and sales department's relevant information just mentioned.

Basically there are two different techniques for providing the peers (departments) with the required data. Contrary to the commonly used method querying the data sources actively, our approach uses replication of data and schema, initiated by the data source. Referring to our example, the executive board gets data updates whenever changes occur in the sales and/or product engineering databases rather than having to request for updated data items continuously.



**Fig. 1.** DÍGAME Architecture

Our DÍGAME architecture (Figure 1) consists of the following components:

**Autonomous Component Databases:** As already mentioned our architecture is designed for integrating data from across autonomous data sources. The peers involved are linked to an information grid, retaining their local autonomy completely. According to the 3-level-architecture [4] and the architecture for loosely coupled multidatabases [3], each component database consists, besides the internal schema, of a conceptual and several external schemas. The conceptual schema (CS) comprises the locally maintained private schema (PS) and a couple of schemas imported from other peers (IS), managed by a wrapper component. Local applications (A) access the data

via external schemas (ES), which are derived from the whole conceptual schema, providing solely read-only access to all imported schemas. The grid infrastructure does not include a global view over the integrated data, but every peer maintains its own integrated schema. Due to the absence of a global view, we have ideal conditions for individual integrations on each peer.

**Wrapper:** The core of our dynamic information grid is the wrapper component. As part of the middleware, it is responsible for negotiating and establishing communication and exchanging data between the component databases. Therefore it maintains a *repository* containing all the meta data accumulated, particularly a copy of the import schemas mentioned above and export schemas (XS) based exclusively on the private schema. Of course corresponding schema information has to be stored only if data is imported or exported. Thus the participation level corresponds directly to the amount of schema information the wrapper has to manage. In fact each import schema matches an export schema, offered by one of the remaining peers. In addition to the repository, two more modules have to be implemented inside the wrapper. A *publishing unit* is used for transmitting information. Earlier research proposes several mechanisms helping a wrapper to identify data modifications [5, 6]. If there are triggers of underlying database systems available, they should be used. The counterpart of the publisher is a *subscribing unit*, which receives incoming information. Both units contain a *negotiator*, which sets up a communication channel, used by a *data handler*, to exchange data in a standardized format (e.g. RDF).

### 2.3 Characteristics

Our architecture combines the advantages of established concepts known not only in the database field, but also in related research areas, like grid or P2P computing. The combination of the established *Three Schema Architecture* [4] and that of loosely coupled multidatabases [3] with the achievements of the more recent field of P2P data management [7] provides a promising framework for an enterprise information platform. We are thus able to apply the flexible interconnectivity of P2P systems to multidatabases, including not only relational databases, but a loosely coupled federation of virtually any kind of data source. Although we focus in this paper on relational database systems, our architecture can be adapted to X.500 directory services or even file systems by adjusting the wrapper component.

In fact, the data replication on each subscriber database provides a couple of advantages, according to distributed databases systems [8]. As the peers serve all their local applications with the data required, there is no need for these to query remote data sources, leading to an increase of performance. Furthermore, a temporary network blackout can be bridged without being noticed by the applications. Comparable to distributed databases, in scenarios with more data queried than modified, even a significant reduction of network traffic can be

obtained. Of course a replication entails the redundant storage of data items, but this disadvantage is neglectable due to the rapid decrease of storage costs.

To ensure a high level of data quality, the data can only be modified by the data owner. Information sharing between autonomous data sources is realized without loss of data ownership and autonomy on each peer, leading to a higher quality of data [9].

To guarantee both, the correctness and up-to-dateness of the data, each single modification can be propagated by pushing it to the subscriber databases. Hence each peer is able to provide, a running network environment supposed, up-to-date data to its applications at any time. Due to the push-based characteristics of DÍGAME updates may be lost if a communication failure caused by a network or computer breakdown occurs. In this case there are basically two possibilities to re-synchronize the data. Either the publishing peer repeats the lost update propagations or the subscribing peer itself demands for these data and schema modifications. Since the first option enforces the data source to keep a complete track on the success of every propagation to each subscribers, we prefer to integrate a pull-based fallback mechanism into our architecture. This means that the subscribing peer has to search for lost data and schema updates by itself. The concrete definition of this functionality is part of future work.

### 3 Related Work

Simultaneously to the first generation of grid computing in the mid 1990s [10] some efforts arised to use distributed resources for information retrieval. Although the *Information Grid* of Rao et al. [11] is focused on giving an integrative user interface for distributed information, this approach can be seen as an early forerunner of the so called *Data Grid* [12], a specialization and extension of grid computing. Its intention is to create an architecture of integrated heterogeneous technologies in a coordinated fashion. Though we admit that a global metadata repository as proposed by Chervenak et al. would simplify many of the challenges, we abstain from that that effort of re-centralization, as it brings many difficulties about: every schema change has to be replicated to the global schema directory. The effect is a single point of failure, exactly the opposite of what we wanted to construct. We thus prefer to keep the databases as they are: autonomous, loosely coupled, and without a single point of failure.

With the raise of filesharing systems like Napster or Gnutella [13] the database community started to seriously adopt the idea of P2P systems to the formerly known loosely coupled database systems. Contrary to the data grid, P2P database systems do not have a global control in form of a global registry, global services, or a global resource management, but multiple databases with overlapping and inconsistent data. These P2P databases resemble heterogeneous and distributed databases, also known as *multidatabases* [14]. Currently the database community makes a great effort in investigating P2P databases. Particularly the *Piazza* [7] project is worth mentioning, where a P2P system is built up with the techniques of the *Semantic Web* [15] with local point-to-point

data translations, rather than mapping to common mediated schemas or ontologies. Contrary to Halevy et al., we deal mainly with relational data and do not have a global schema, as every peer may have its own import-/export-schema combination. For a more general glimpse on data mappings in P2P systems see [16].

Our strategy allows data to be exchanged among distributed databases connected through a lazy network. This means, that although a running network may not be guaranteed and thus some data broadcasts may be lost, the system heals itself. This challenge resembles the problems known from environments with mobile databases. Current research covers synchronous mobile client synchronization, i.e. data changes are propagated periodically and not just in time of the data change. In contrast to the broadcast disks, we ensure in our model, that data is only broadcasted to the clients when changes occur, unless the communication between both peers crashes. Hence our approach resembles a *push-based* system with a *pull-based* fallback, similar to [17] with the major difference that our approach is not based on broadcast disks, but on a push-based replication strategy also found in mobile clients like [18], resembling the software engineering's *Observer-Pattern* [19]. This pattern gives us a prototype of how to notify all interested databases about data updates [18]. As a result, communication is only started, if a data update has occurred and a database is interested. In consequence, data broadcasts are minimized.

Following the argumentation in [20] and [12] our model provides Single-Master Replication, the only guarantor for data stability and clear defined data flows.

Most of the research on active multidatabases has been done concerning global integrity. Chawathe et al. [21] propose a toolkit for constraint management in loosely coupled systems. To mention is also the idea of Gupta and Widom to optimize the testing of global constraints by local verification [22]. Conrad and Türker [6] sketch a more general architecture for an active federated database system. They extend a multidatabase system by ECA-Rules to preserve consistency. A main challenge hereby is to detect local events, especially schema and data modifications, which is commonly done by a software module for each data source, i.e. a monitor or wrapper component. Basically two approaches are therefore proposed: Conrad and Türker use the event detection ability of the underlying subsystem, while Blanco et al. [5] use the operating system to signal schema modifications by directly observing changes to the data(base) files.

## 4 Conclusion and Future Work

We have presented in this paper an architecture for a dynamic information grid in an active multidatabase environment, suitable for sharing information across autonomous and heterogeneous data sources. Our loosely coupled federation enriched with P2P and grid computing concepts, enables collaborative work preserving local autonomy. Data and schema modifications are actively propagated

to the clients after they have subscribed to the information offered by the data source.

For a complete implementation of the DÍGAME architecture, there are still some challenges to be taken. In the next step of the project we will focus on the detailed specification of the wrapper component, particularly on the negotiator and the data handler, for being able to establish communication between isolated peers. Therefore we have to specify a communication protocol and a data and schema exchange format.

Due to its characteristics DÍGAME provides a sophisticated infrastructure for a diversified application field, including e-business, e-science or e-health, initiating the next generation of collaborative work.

## References

1. Litwin, W., Abdellatif, A.: Multidatabase Interoperability. *Computer* **19** (1986) 10–18
2. Mullen, J.G., Elmagarmid, A.K., Kim, W., Sharif-Askary, J.: On the Impossibility of Atomic Commitment in Multidatabase Systems. In: *Proceedings of the 2nd International Conference on System Integration*, Morristown, New Jersey, IEEE Computer Society Press (1992) 625–634
3. Heimbigner, D., McLeod, D.: A Federated Architecture for Information Management. *ACM Transactions on Information Systems (TOIS)* **3** (1985) 253–278
4. Burns, T., Fong, E.N., Jefferson, D., Knox, R., Mark, L., Reedy, C., Reich, L., Roussopoulos, N., Truszkowski, W.: Reference model for dbms standardization, database architecture framework task group (daftg) of the ansi/x3/sparc database system study group. *SIGMOD Record* **15** (1986) 19–58
5. Blanco, J.M., Illarramendi, A., Pérez, J.M., Goñi, A.: Making a Federated Database System Active. In: *Database and Expert Systems Applications*, A.M. Tjoa and I. Ramos (eds.), Springer Verlag, ISBN 3-211-82400-6. (1992) 345–351
6. Türker, C., Conrad, S.: Towards Maintaining Integrity of Federated Databases. In: *Data Management Systems, Proc. of the 3rd Int. Workshop on Information Technology, BIWIT'97*, July 2–4, 1997, Biarritz, France, Los Alamitos, CA, IEEE Computer Society Press (1997) 93–100
7. Halevy, A.Y., Ives, Z.G., Mork, P., Tatarinov, I.: Piazza: Data Management Infrastructure for Semantic Web Applications. In: *Proceedings of the twelfth international conference on World Wide Web*, Budapest, Hungary (2003) 556–567
8. Ceri, S., Pelagatti, G.: *Distributed databases principles and systems*. McGraw-Hill, Inc. (1984)
9. van Alstyne, M., Brynjolfsson, E., Madnick, S.: Why not one big database?: principles for data ownership. *Decision Support Systems* **15** (1995) 267–284
10. Roure, D.D., Baker, M.A., Jennings, N.R., Shadbolt, N.R.: The Evolution of the Grid. In Berman, F., Fox, G., Hey, T., eds.: *Grid Computing: Making the Global Infrastructure a Reality*. John Wiley & Sons Inc., New York (2003) 65–100
11. Rao, R., Card, S.K., Jellinek, H.D., Mackinlay, J.D., Robertson, G.G.: The Information Grid: A Framework for Information Retrieval and Retrieval-Centered Applications. In: *Proc. of the 5th Annual Symposium on User Interface Software and Technology (UIST'92)*, Monterey, CA (1992) 23–32

12. Chervenak, A., Foster, I., Kesselman, C., Salisbury, C., Tuecke, S.: The Data Grid: Towards an Architecture for the Distributed Management and Analysis of Large Scientific Datasets. *Journal of Network and Computer Applications* **23** (2000) 187–200
13. Carlsson, B., Gustavsson, R.: The Rise and Fall of Napster - An Evolutionary Approach. In: AMT 2001, Proceedings of the 6th International Computer Science Conference - Active Media Technology. Volume 2252 of Lecture Notes in Computer Science., Hong Kong, China, Springer (2001) 347–354
14. Bernstein, P.A., Giunchiglia, F., Kementsietsidis, A., Mylopoulos, J., Serafini, L., Zailhaye, I.: Data management for peer-to-peer computing: A vision. In: Proc. of the Fifth International Workshop on the Web and Databases, WebDB 2002, Madison, WI (2002)
15. Berners-Lee, T., Hendler, J., Lassila, O.: The semantic web. *Scientific American* (2001)
16. Kementsietsidis, A., Arenas, M., Miller, R.J.: Mapping Data in Peer-to-Peer Systems: Semantics and Algorithmic Issues. In: Proceedings of the 2003 ACM SIGMOD international conference on Management of data, San Diego, CA, ACM Press (2003) 325–336
17. Acharya, S., Franklin, M., Zdonik, S.: Balancing push and pull for data broadcast. In: Proceedings of the 1997 ACM SIGMOD international conference on Management of data, Tucson, Arizona, ACM Press (1997) 183–194
18. Hara, T.: Cooperative caching by mobile clients in push-based information systems. In: Proceedings of the eleventh international conference on Information and knowledge management, McLean, Virginia, USA (2002) 186–193
19. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley Professional Computing Series. Addison-Wesley Publishing Company, New York, NY (1995)
20. Gray, J., Helland, P., O’Neil, P., Shasha, D.: The Dangers of Replication and a Solution. In: Proceedings of the 1996 ACM SIGMOD international conference on Management of data, Montreal, Canada, ACM Press (1996) 173–182
21. Chawathe, S., Garcia-Molina, H., Widom, J.: A Toolkit For Constraint Management In Heterogeneous Information Systems. In: Proceedings of the International Conference on Data Engineering, New Orleans, Louisiana (1996) 56–65
22. Gupta, A., Widom, J.: Local Verification of Global Integrity Constraints in Distributed Databases. In: Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data SIGMOD’93, Washington, DC (1993) 49–58

# Typische Integrationsszenarien und deren Unterstützung durch Web Services und andere Technologien

Cornelia Boles, Jörg Friebe, Till Luhmann

OSC – IM Systems AG  
Industriestr. 11  
26121 Oldenburg

[cornelia.boles@osc-ims.de](mailto:cornelia.boles@osc-ims.de), [joerg.friebe@osc-ims.de](mailto:joerg.friebe@osc-ims.de), [till.luhmann@osc-ims.de](mailto:till.luhmann@osc-ims.de)

**Abstract.** Die Integration existierender und neu einzuführender Anwendungen ist heute eine Kernaufgabe im Bereich der IT. Um für das jeweilige Integrationsszenario die geeignete Integrationstechnologie auswählen zu können, müssen die Integrationsszenarien und die daraus resultierenden Anforderungen an die Integrationstechnologien genauer untersucht werden. Der vorliegende Artikel beschreibt nach einer einführenden Betrachtung der möglichen Integrationsebenen die charakteristischen Merkmale von Integrationsszenarien. Im Anschluss daran werden die wichtigsten Integrationstechnologien kurz vorgestellt und bezüglich ihrer Eignung für die einzelnen Merkmalsausprägungen der Integrationsszenarien bewertet. Der Artikel schließt mit einer Zusammenfassung und einem Ausblick.

## 1 Einleitung

Das Thema Web Services wird derzeit in vielen Publikationen und auf großen Konferenzen behandelt. Doch Web Services sind längst kein reines Forschungsthema mehr. Vielmehr werden sie vermehrt in Unternehmen eingesetzt, in erster Linie zur Integration von Anwendungen. Die Vielzahl der im Umfeld von Web Services entstehenden Standards (z.B. zu den Themen Sicherheit, Transaktionen oder Web Services Choreographien) zeigt, dass die Technologie heute einerseits noch nicht ganz ausgereift ist, sie aber andererseits als Basis angesehen werden kann, auf der aufbauend weitere Technologien entstehen werden. Die Kernaspekte von Web Services sind jedoch soweit standardisiert, dass sie sich auch in Produktivumgebungen einsetzen lassen.

Im Gebiet der Anwendungsintegration stellt sich die Frage, wie sich Web Services im Vergleich zu etablierten Integrationstechnologien und Werkzeugen (z.B. aus dem Enterprise Application Integration Umfeld) positionieren. Um diese Frage beantworten zu können, werden im vorliegenden Artikel Integrationsszenarien genauer untersucht und anhand ihrer charakteristischen Merkmale klassifiziert.

Der Artikel gliedert sich dabei wie folgt: Nach einer grundlegenden Betrachtung der Ebenen, auf denen Softwaresysteme integriert werden können, werden in die charakteristischen Merkmale von Integrationsszenarien diskutiert. Der letzte Abschnitt stellt bestehende Integrationstechnologien vor und geht darauf ein, für welche

Integrationsmerkmale die verschiedenen Technologien geeignet sind. Der Artikel schließt mit einer Zusammenfassung und einem Ausblick.

## **2 Integrationsebenen**

Die Integration von Softwaresystemen in der Informationsverarbeitung kann mindestens auf den Ebenen der Daten, Anwendungen und Prozesse betrachtet werden [BFGH02]. Die dabei relevanten Aspekte werden im Folgenden dargelegt.

### **Datenintegration**

Grundlegende Techniken zur Integration von Informationssystemen sind auf der Ebene der Datenbestände bzw. auf Basis der Schemata zugrunde liegender Datenbanken angesiedelt. Ziel ist fast immer die Sicherstellung anwendungsübergreifender Konsistenz der Daten. Eine erste naheliegende Lösung sieht die Einführung eines unternehmensweit integrierten Datenschemas vor. Die Idee ist hier, nur eine Datenhaltungskomponente und nur ein Datenschema zu verwenden und Anwendungen so zu entwickeln, dass sie auf dieses Datenschema aufsetzen. Somit können die Konsistenzsicherungsmechanismen des DBMS genutzt werden. Dieser Ansatz ist jedoch nur sinnvoll verwendbar, wenn Anwendungen neu entwickelt bzw. reimplementiert werden sollen oder können. Die oft in Unternehmen vorzufindende heterogene Systemlandschaft macht andere Ansätze erforderlich.

Zur Datenintegration wird in heterogenen Systemlandschaften häufig Datenzugriffsmiddleware eingesetzt. Sie ermöglicht neben einer für Anwendungen transparenten Einbindung externer Daten den Erhalt vorhandener Zuständigkeiten für Datenpflege und -verwaltung sowie die Wahrung der Autonomie existierender Anwendungen. Verwendete Technologien sind hier Open Database Connectivity (ODBC) und deren Nachfolgertechnologien wie OLEDB und ADO.net sowie die Java Database Connectivity (JDBC) und deren weiterführende Entwicklungen wie Entity Beans im J2EE-Umfeld. Die Integration der Schemata erfolgt hier somit lediglich auf logischer Ebene. Als Voraussetzung für die Integration externer Datenbestände müssen Systemhersteller die Semantik der Datenschemata offen legen bzw. den direkten Zugriff auf den Datenbestand freischalten. Ist dieser Zugriff nur über spezielle Application Programming Interfaces (APIs) möglich, so sind Ansätze, die nur auf Mechanismen beteiligter (relationaler) Datenbankmanagementsysteme (DBMS) basieren, nicht anwendbar.

### **Anwendungsintegration**

Neben der Integration von Schemata und Daten verschiedener Anwendungen ist jedoch auch die Integration der Anwendungen selbst wichtig. Ziel ist dabei einerseits die oben genannte anwendungsübergreifende Konsistenz der Daten. Andererseits wird die möglichst weitgehende Wiederverwendung und Nutzung von vorhandener Funktionalität aus den integrierten Systemen (funktionale Integration) sowie eine ein-

heitliche Benutzungsoberfläche (GUI-Integration) angestrebt. Als Basis für die Systemkopplung dienen Integrationsplattformen, welche die erforderlichen Dienste bereitstellen. Derzeit bieten sich mit .net [MSOFT03] und J2EE [Mon02] zwei Integrationsplattformen an, die die notwendige Infrastruktur bereitstellen. Mit Einschränkungen kann auch CORBA [OMG02] hier genannt werden, doch insbesondere in Bezug auf GUI-Integration (Nutzung von Web-Frontends) sowie Verfügbarkeit und Unterstützung der Schnittstellen durch Anwendungen und Komponenten (Einbettung von Visual Basic bzw. zahlreicher Java-Pakete und APIs) bieten J2EE und .net hier Vorteile.

Oft kommt bei der Anwendungsintegration Message-Oriented Middleware [Bern96] zum Einsatz. J2EE bietet hier das Konzept der Message Queues, Publish/Subscribe-Mechanismen und Message-Driven Beans. Die damit realisierbaren Architekturen verbinden funktionale Integration mit den Vorteilen asynchroner Kommunikation (Send and Forget) und erlauben eine lose Kopplung der beteiligten Anwendungen, denn oft besteht die Anforderung, dass Systeme durch Downzeiten angekoppelter Systeme nicht in ihrer eigenen Verfügbarkeit beeinträchtigt werden dürfen. Das Nachrichtenformat ist typischerweise XML-basiert, um die Einbindung weiterer Systeme zu erleichtern. Die funktionale Integration allein reicht jedoch nicht aus, da hierdurch zwar Funktionalität wiederverwendet werden kann, nicht aber Dialogelemente, die zur Parametrisierung bzw. Präsentation der Funktionalität nötig sind. Aus diesem Grund wird die Integration der Anwendungen auch auf der Ebene der Benutzungsoberflächen durchgeführt. Hierzu sind mehrere Wege denkbar. Der klassische Weg führt zur Integration von GUI-Controls, wie sie als Microsoft Foundation Classes, Java AWT oder Swing bekannt sind [Sun02]. Teilweise lassen sie sich auch im Java-Applet-Umfeld nutzen. Ist jedoch der Einsatz im Internet erforderlich (z.B. im B2B-Umfeld oder im Bereich Customer Self Service), so sind schlanke, nahezu überall lauffähige Frontends erforderlich, die auf den Client-Rechnern keine Arbeiten (Installation, Administration) erfordern. Die Wahl fällt hier auf HTML-Clients, die z.B. mit den Technologien ASP.net bzw. JSP realisiert werden können.

### **Prozessintegration**

Daten- und Anwendungsintegration reichen nicht aus, wenn komplexe Geschäftsprozesse unterstützt werden müssen, die zudem einem ständigen Wandel unterliegen. In diesem Fall muss es mit einfachen Mitteln möglich sein, die Geschäftsprozesse um Aktivitäten zu erweitern oder anzupassen. Hierzu bedarf es jedoch einer übergeordneten Steuerungsebene, mit der anwendungsübergreifende Geschäftsprozesse realisiert werden können. Um dies umzusetzen, werden Konzepte aus den Bereichen Workflowmanagement (WFM) und Enterprise Application Integration (EAI) bemüht. Hierbei werden die Workflow-Schemadaten, Workflow-Instanzdaten sowie Akteure und Aktionen zentral in einem Prozesssteuerungssystem verwaltet. Je nach Art der Prozesse kann dies ein Workflowmanagementsystem (WfMS) oder eine EAI-Engine sein. Im Mittelpunkt der Prozessintegration stehen die betriebswirtschaftlichen und auf den Kunden ausgerichteten Geschäftsprozesse. Ziel der Prozessintegration ist die vollständige elektronische Bearbeitung aller kundenorientierten Prozesse innerhalb des Unternehmens. Zentrale Aspekte sind hier die automatischen Aktivitäten und die

Arbeitskorb-Metapher. Während erstere völlig ohne Eingriff des Anwenders durch das System gesteuert werden (durch asynchrone Nachrichtenkommunikation, in Einzelfällen auch durch synchronen Aufruf von z.B. Session Beans), stellt der Arbeitskorb die zentrale Schnittstelle zwischen Prozess und Benutzer dar. Auch hier steht die asynchrone Bearbeitung im Vordergrund. Wird z.B. in einem Geschäftsprozess nach Ausführung von automatischen Aktivitäten eine Bearbeitung durch einen Anwender notwendig, so erhält dieser (oder seine Rolle) im Arbeitskorb eine entsprechende manuelle Aktivität zugeordnet. Die Aktivität ist z.B. anhand von Web-Formularen durchzuführen, die vom System dem Anwender prozessgesteuert vorgelegt werden. Nach der Bearbeitung übernimmt das System die weitere Steuerung des Prozesses.

### **3 Charakteristika eines Integrations szenarios**

Betrachtet man die in konkreten Projekten auftretenden Integrations szenarien genauer, so stellt man schnell fest, dass sie sich oftmals nicht genau einer Integrationsebene zuordnen lassen. Vielmehr wachsen die Integrationsebenen zusammen, so dass die Integration auf mehreren Ebenen erfolgt. Damit reicht die Klassifikation nach Integrationsebenen nicht aus, um Integrations szenarien zu charakterisieren und die für das jeweilige Szenario geeignete Integrationstechnologie auszuwählen.

Um dennoch die Auswahl der geeigneten Technologie unterstützen zu können, haben die Autoren die charakteristischen Merkmale von Integrations szenarien analysiert. Dabei haben wurden sechs Merkmale identifiziert: Automatisierungsgrad und Benutzerinteraktion, Strukturiertheit der Daten, Dynamik der Daten, Anzahl der zu integrierenden Systeme, Kommunikationsparadigma und Status und Zustandshaltung, die im Folgenden genauer erläutert werden.

#### **Automatisierungsgrad und Benutzerinteraktion**

Zwei fast immer in Integrations szenarien relevante, zudem komplementäre Aspekte sind der Grad der Automatisierbarkeit von Prozessschritten und der Grad der benötigten Benutzerinteraktion. Ist der Grad der Automatisierbarkeit hoch, so können die einzelnen Prozessschritte bzw. Aktivitäten ohne Interaktion durch einen Bearbeiter durchgeführt werden. Ziel der Geschäftsprozessoptimierung ist es typischerweise, diesen Grad zu erhöhen. Ein hoher Automatisierungsgrad bedeutet zudem, dass Aspekte wie Benutzerverwaltung, Rollen, Authentifizierung weniger relevant sind. Einzelne Aktivitäten können vom System direkt ohne Rückkopplung durch den Anwender durchgeführt werden. Dies bedeutet auch, dass die Lokalisation der Aktivität unkritisch ist, und z.B. von Loadbalancing-Strategien bestimmt werden kann (serverbasierte Durchführung automatischer Aktivitäten) und dass synchrone Kopplung nicht notwendig ist (da kein Benutzer unmittelbar auf Rückmeldung wartet). Ein hoher Automatisierungsgrad erfordert effiziente Konzepte für Interprozesskommunikation, Multithreading und Schnittstellen zu zu steuernden Komponenten (APIs).

Dem entgegen steht der Grad der notwendigen Benutzerinteraktion bei der Bearbeitung von Geschäftsprozessen. Notwendige Benutzerinteraktionen reduzieren den

Grad der Automatisierbarkeit und sollten minimiert werden. Derartige Interaktionen können jedoch vielfältig sein, z.B. Synchronisationspunkte innerhalb weitgehend automatisierter Prozesse, an denen der Anwender automatisch erstellte Berechnungsergebnisse prüfen muss, bevor die nächste automatische Aktivität gestartet wird, Erfassung von zusätzlichen Daten über Bildschirmformulare oder manuelle Bearbeitung von Dokumenten (z.B. Textdokumente oder Grafiken). Benutzerinteraktionen erfordern einen Mechanismus, über den Aktivitäten einzelnen Anwendern oder Rollen zugeordnet werden können (Arbeitskorb-Metapher). Dieser Mechanismus wird insbesondere zur asynchronen Kopplung genutzt. Zur Einbindung von Formularen ist ein entsprechendes Framework nötig. Weiterhin sind clientseitige Schnittstellen zu Anwendungen notwendig, die zur Unterstützung bestimmter Aktivitäten (z.B. Dokumentbearbeitung) notwendig sind.

Der Grad der Automatisierbarkeit und Benutzerinteraktion variiert und ist vom Geschäftsprozess abhängig. In den seltensten Fällen liegen in Unternehmen die Extremsituationen (sehr hohe Automatisierung, sehr geringe Benutzerinteraktion bzw. sehr geringe Automatisierung, sehr hohe Benutzerinteraktion) vor.

### **Strukturiertheit der Daten**

Ein wichtiger Aspekt bei der Integration ist die Struktur bzw. der Strukturierungsgrad der zu integrierenden Daten. In Unternehmen ist dabei typischerweise eine große Bandbreite an Daten anzutreffen, die in Geschäftsprozessen relevant sind. Zur Optimierung der Geschäftsprozesse ist es sinnvoll, den Anteil strukturierter Daten zu erhöhen und den Anteil an unstrukturierten Daten zu verringern, denn strukturierte Daten erlauben einen höheren Automatisierungsgrad, während unstrukturierte Daten fast immer erweiterte Benutzerinteraktionen, z.B. Interpretation, erfordern.

Beim Austausch auf Dateiebene sind die Daten häufig rudimentär formatiert, z.B. in Form kommaseparierter Werte. Diese lassen sich innerhalb einer Integrationsplattform durch File-Adapter einlesen und weiterverarbeiten. Ist die Struktur der Daten selbst komplex, wie z.B. bei EDI-Formaten, so gibt es heute zu einer XML-Darstellung praktisch keine Alternative, weil der individuelle Entwicklungsaufwand für das Einlesen und Parsen der Daten dadurch entfällt. Handelt es sich hingegen um Dokumente in proprietären Formaten, wie z.B. Word-Dateien, so werden die Daten in der Regel nur als Ganzes weitergeleitet oder archiviert. Zwar existieren inhaltliche Zugriffsmöglichkeiten über Objektmodelle, doch die individuellen Entwicklungsaufwände für eine inhaltliche Interpretation der Dokumente sind meist beträchtlich, so dass hierfür einfachere Mechanismen eingesetzt werden. Die letzte Gruppe der zu verarbeitenden Daten bilden die unstrukturierten Dokumente, wie z.B. Bilddateien, Faxe oder eingescannte Schriftstücke.

### **Dynamik der Daten**

Als Datenbasis einer Integrationslösung kann im Prinzip die gemeinsame Datenbasis aller angeschlossenen Anwendungen angesehen werden. Ziel beim Betrieb einer Integrationsplattform sollte es sein, die Gesamtheit der Daten in einem konsistenten

Zustand zu halten. Da sich bei einer Vielzahl von angeschlossenen Systemen Datenredundanzen nicht vermeiden lassen, sind Abgleichsmechanismen zu implementieren. Hierbei muss sorgfältig zwischen der Häufigkeit von Datenänderungen, der Notwendigkeit einer Verfügbarkeit aktueller Daten in einzelnen Systemen und dem beim Datenabgleich entstehenden Kommunikationsaufwand abgewägt werden.

Bei bestimmten hochdynamischen Daten ist die Aktualität das entscheidende Wertmerkmal. Hierzu zählten Börsendaten, aber auch Wetterdaten. Entscheidend ist hier, dass die Daten nur in dem Moment der Messung bzw. Aufnahme wirklich aktuell sind. Daher müssen die Daten möglichst zeitnah beim Interessenten zur Wahrnehmung gelangen. Zum Beispiel werden hereinkommende Daten als Nachrichten entgegengenommen und sofort auf die Zielsysteme zum Abruf verteilt.

In anderen Fällen können zum automatisierten Datenabgleich mit mehreren Zielsystemen Publish-Subscribe-Mechanismen eingesetzt werden. Sofern die Änderungen plattformseitig gesammelt werden und der Abgleich nur selten stattfindet, ist es sinnvoll, die in einem System verfügbaren Daten mit einem Aktualitätszeitstempel zu versehen. Betriebsintern kann dem Bearbeiter die Möglichkeit eingeräumt werden, den Datenabgleich bei Bedarf manuell anzustoßen. Gängige EAI-Plattformen bieten hierfür entsprechende GUIs an.

### **Anzahl der zu integrierenden Systeme**

Soll eine zentrale Integrationsplattform eingesetzt werden, um die Anzahl individueller Systemschnittstellen (Punkt-zu-Punkt-Verbindungen) zu reduzieren, so fällt auf, dass bei wenigen Systemen häufig mehr Schnittstellen von und zu der Integrationsplattform zu implementieren sind als ursprünglich Punkt-zu-Punkt-Verbindungen existieren. In der Regel ist nämlich nur ein Bruchteil aller möglichen Punkt-zu-Punkt-Schnittstellen sinnvoll und daher auch realisiert, wohl aber müssen als Voraussetzung für die Integration Kopplungen aller Systeme zur zentralen Plattform erstellt werden. Das Argument der Aufwandsreduktion ist somit erst ab einer größeren Anzahl von Systemen wirksam. Wird hingegen auf die höhere Qualität der Systemanbindungen und die Mehrwertdienste einer Integrationsplattform Wert gelegt, so tritt das Aufwandsargument in den Hintergrund. Daher kann eine plattformgestützte Integrationslösung bereits bei nur zwei Systemen sinnvoll sein.

Werden sehr viele Systeme integriert, so muss beachtet werden, dass mit der Anzahl der Systeme sich in der Regel auch die Technologien der Adapter immer stärker diversifizieren. Setzt die Plattform schwerpunktmäßig auf individuelle Adapterentwicklung, so können hier erhebliche Aufwände schlummern. Im Vorteil sind hier Plattformen, die bereits einen großen Satz von konfigurierbaren technologiespezifischen Adaptern mitbringen. Alternativ dazu bietet es sich an, Adapter auf Basis von Web Services zu realisieren. Hierdurch schafft man eine Anwendungskonnektivität, die unabhängig von der gewählten Integrationsplattform ist.

Sind die Systeme auf verschiedene, durch Firewalls getrennte Netzwerke verteilt, so kann die Integrationsplattform in jedem Netzwerk Kommunikationsserver platzieren, die über freigeschaltete Ports kommunizieren und damit eine Art Backbone für die Prozesskommunikation darstellen. Dies erlaubt es der Plattform theoretisch, Prozesse zentral administrierbar und überwachbar zu halten. Eine Alternative

hierzu stellt ein zentraler Server dar, der über ein VPN direkt mit den verschiedenen Systemen kommuniziert.

### **Kommunikationsparadigma**

Eine besonders lose Form der Kommunikation ist der Export und Import von Austauschdateien: Die Quellanwendung schreibt ihre Daten in ein File in einem Austauschverzeichnis, welches von der Zielanwendung oder von einem Importskript im Batchbetrieb regelmäßig geprüft wird („push and poll“). Gerade Altsysteme bieten häufig nur diese Art des Datenaustausches an. Auch beim Einsatz einer Integrationsplattform muss die Ankopplung dieser Systeme asynchron über individuell konfigurierte File-Adapter erfolgen. Insbesondere die koordinierte Verteilung der Daten auf mehrere Zielsysteme (z.B. beim Stammdatenabgleich) wird durch die Plattform besser unterstützt als z. B. auf der Ebene der Shellskripte. Als weitere Form der asynchronen Kommunikation z.B. mit externen Systemen werden E-Mails eingesetzt. Eingehende E-Mails werden über spezielle Adapter interpretiert und weiterverarbeitet, ebenso werden E-Mails plattformseitig inhaltlich generiert und versandt.

Verfügt eine Anwendung über ein Remote API oder handelt es sich um eine datenbankgestützte Anwendung, so stehen verschiedene Mechanismen der synchronen Kommunikation offen: Zunächst kann die Anwendung über einen individuell zu entwickelnden bzw. proprietären Systemadapter der Integrationsplattform angesprochen werden. Alternativ dazu bietet sich eine Systemanbindung mittels Web Services an: Für verschiedene Programmiersprachen (z.B. Java und .net-Sprachen) stehen heute mächtige Tools zur Verfügung, mit denen aus einem Remote API anwendungsspezifische Web Service Schnittstellen inkl. WSDL-Beschreibungen generiert werden können. Bei einer datenbankgestützten Anwendung ohne eigenes API ergibt sich als weitere synchrone Kommunikationsmöglichkeit der pragmatische Weg des direkten Datenbankzugriffs. Der zugehörige individuell konfigurierte Datenbankadapter kann seine Funktionsschnittstelle plattformseitig z.B. über ein Web Service „Facade“ anbieten.

Mit der Anzahl der integrierten Systeme steigt auch der Kommunikationsaufwand innerhalb der Plattform an. Eine Methode, um die Kommunikation zu reduzieren, ist der von Java her bekannte Publish-Subscribe-Mechanismus, der auch von heutigen Integrationsplattformen unterstützt wird. Dabei melden sich nur „interessierte“ Anwendungen beim Verteiler bestimmter, von der Plattform definierter Nachrichten an. Nützlich ist dieses Verfahren z.B. bei komplexeren Koordinierungsaufgaben, wie dem oben erwähnten Stammdatenabgleich.

### **Status und Zustandshaltung**

Der Bearbeitungsstand von Geschäftsprozessen wird klassischerweise manuell über Auftrags- oder Laufzettel erfasst, die von Bearbeitungsinstanz zu Bearbeitungsinstanz weitergereicht werden. Werden solche Vorgänge automatisiert und mittels einer Integrationsplattform gesteuert, so sollten auch die Zustände der einzelnen Prozessinstanzen über diese Plattform fortgeschrieben werden. Hierbei können drei Arten von

Zustandsinformation unterschieden werden: Zunächst ist für jede Prozessinstanz ein globaler Prozesszustand (z.B. „aktiv“) definiert. Ergänzt wird dieser durch ein Protokoll der im Prozessablauf durchgeführten Aktivitäten und der beteiligten Akteure (Systeme, Bearbeiter), d.h. durch eine Beschreibung des Weges, auf dem der aktuelle Prozesszustand erreicht wurde (Historienfortschreibung). Hinzu kommen prozessspezifische Informationen, für welche das System die Möglichkeit einer individuellen Festlegung anbieten muss. Sämtliche Zustandsinformationen sollten durch das System automatisch instanziiert, initialisiert und aktualisiert werden.

Ein weiterer Faktor ist die Persistierung der Statusinformation, ohne die z.B. keine langlaufenden Geschäftsprozesse abgewickelt werden könnten (im J2EE-Umfeld wird diese Aufgabe standardmäßig von Stateful Session-Beans erfüllt). Sinnvoll ist ferner, dass ein Integrationssystem bei Prozessänderungen noch aktive alte und bereits gestartete neue Prozessinstanzen parallel betreiben kann. Hierfür muss das System die gleichzeitige Verwaltung der individuellen Zustandsinformationen beider Prozessversionen unterstützen.

## 4 Integrationstechnologien

Die Entscheidung für eine konkrete Integrationstechnologie ist abhängig von den charakteristischen Merkmalen des Integrationsszenarios. Prinzipiell lassen sich die Integrationstechnologien in die Gruppen „klassische“ EAI-Werkzeuge, Middleware-Technologien, Web Services und Web Services in Kombination mit Business Process Modeling Languages unterteilen. Tabelle 1 gibt einen Überblick über die verschiedenen Integrationstechnologien und zeigt, welche Integrationscharakteristika die einzelnen Technologien unterstützen. Zum Vergleich wird in der Tabelle zusätzlich die Punkt-zu-Punkt-Integration den übrigen Integrationstechnologien gegenübergestellt.

	<b>EAI- Werk- zeuge</b>	<b>WfMS</b>	<b>Middle- ware</b>	<b>Web Services</b>	<b>Web Services + BPML</b>	<b>Punkt- zu- Punkt</b>
Hoher Automati- sierungsgrad	++	+	-	-	++	+ <sup>1</sup>
Viele Benutzerin- teraktionen	o	++	-	-	+	+ <sup>1</sup>
Unterstützung für strukturierte Da- ten	++	o	++	++	++	++
Unterstützung für unstrukturierte Daten	o	++	o	o	+	o
Eignung für Real Time Szenarien	+	+	++	o	o	++

<sup>1</sup> Bestandteil der beteiligten Anwendungen

Eignung für Batch-Prozesse	++	o	+	o	o	+ <sup>1</sup>
Integration vieler Anwendungen	++	+	++	++	++	--
Synchrone Kommunikation	++	+	++	++	++	+
Asynchrone Kommunikation	++	++	++	o	+	+
Unterstützung von Zustandshaltung	++	++	o	-	+	+ <sup>1</sup>

**Tabelle 1.** Eignung der Integrationstechnologien für die verschiedenen Merkmalsausprägungen der Integrationsszenarien

### **Klassische EAI-Werkzeuge**

EAI-Werkzeuge ermöglichen die Integration existierender Anwendungen auf Daten-, Anwendungs- und Prozessebene. Bei der Integration mittels eines EAI-Werkzeugs ist typischerweise für jede Anwendung, die integriert werden soll, ein Adapter zu erstellen, der die Transformation der Daten und der entfernten Systemaufrufe realisiert. Diese Adapter basieren in der Regel auf proprietären Klassenbibliotheken.

Neben der reinen Transformation der Daten und Systemaufrufe stellen die meisten EAI-Werkzeuge Funktionalität zur Lastverteilung, Ausfallsicherheit und zum Monitoring der EAI-Aktivitäten zur Verfügung. EAI-Werkzeuge eignen sich in erster Linie für vollautomatisierte Arbeitsabläufe, die auf strukturierten Daten arbeiten. Dabei wird der Bearbeitungsstand (Status) der einzelnen Arbeitsabläufe in der Regel vom EAI-Werkzeug registriert und dann über das EAI-Werkzeug überwacht.

Mit der wachsenden Popularität der Web Services haben auch die EAI-Anbieter das Potential dieser Technologie erkannt. So bieten heute viele EAI-Werkzeuganbieter die Integration nicht mehr nur mittels proprietärer Adapter an, sondern ermöglichen es den Integratoren, Anwendungen über Web Services an das EAI-Werkzeug anzubinden.

### **Workflow Management Systeme**

Ebenso wie klassische EAI-Systeme dienen auch WfMS der Automatisierung von Arbeitsabläufen. Der wesentliche Unterschied zwischen diesen beiden Systemklassen liegt jedoch in der Art der Anwendungen, die automatisiert werden sollen. Während bei EAI-Systemen vollautomatisierte Prozesse im Vordergrund stehen, konzentrieren sich WfMS auf Arbeitsabläufe, die Benutzerinteraktionen erfordern. Der Bedarf an Benutzerinteraktionen ist auch dadurch begründet, dass WfMS mit unstrukturierter Daten arbeiten. Um diese Daten geeignet interpretieren und verarbeiten zu können, ist meist der Eingriff eines Menschen erforderlich.

## **Middleware-Technologien**

Middleware-Technologien für sich genommen sind nicht ausreichend für die Integration existierender Anwendungen. Sie bieten jedoch eine Infrastruktur, mittels derer Anwendungen miteinander kommunizieren, Daten ausgetauscht und entfernte Methoden einer anderen Anwendung aufgerufen werden können. Damit eignen sich Middleware-Technologien für eine Integration auf Daten- und/oder Anwendungsebene. Die Prozessintegration wird durch die Middleware-Technologien nur insoweit unterstützt, dass entfernte Systemaufrufe erfolgen können. Wann, wie und in welcher Reihenfolge diese Aufrufe erfolgen, wird von den beteiligten Anwendungen bestimmt. Die Middleware selbst hat von Abhängigkeiten zwischen den einzelnen Systemaufrufen und damit vom Status der Arbeitsabläufe keinerlei Kenntnis.

## **Web Services**

Die Web Services Standards bieten standardisierte Protokolle (SOAP) und Schnittstellenbeschreibungen (WSDL) für das Auffinden (UDDI) und die Nutzung entfernter Dienste. So kann ein Anbieter Anwendungsfunktionalität als Dienst über einen Web Service zur Verfügung stellen, der von beliebigen Nutzern (Personen oder Softwaresystemen) genutzt werden kann. Web Services eignen sich damit für eine lose Kopplung von Softwaresystemen.

Die Web-Services-Technologie erlaubt es nicht, verschiedene aufeinanderfolgende Dienstaufrufe zueinander in Beziehung zu setzen (ausser die beteiligten Anwendungen regeln dies), oder eine Abfolge von verschiedenen Dienstaufrufen festzulegen. Sie befasst sich immer nur mit genau einer Nutzung eines Dienstes.

## **Web Services in Kombination mit Business Process Modeling Languages**

Zur Spezifikation von Geschäftsprozessen sind in den letzten Jahren eine Reihe von Standards wie BPEL4WS, BMPL, XPDL, UML entstanden, wobei BPEL4WS, BMPL und XPDL XML-basierte Sprachen sind [BPML03, BPEL03]. BPEL4WS (Business Process Execution Language for Web Services) ist speziell auf die Koordination von Web Services ausgerichtet. Gesteuert und überwacht wird die Ausführung eines in BPEL4WS spezifizierten Geschäftsprozesses von einer sogenannten „BPEL Engine“. Derzeit ist die Entwicklung der Web Services noch nicht ausgereift. So gibt es noch einige offene Fragen bzgl. der Sicherheit und Transaktionen im Zusammenhang mit Web Services. Diese Probleme wurden jedoch erkannt, und es sind Standards im Entstehen, die sich mit den genannten Fragestellungen befassen. Für die Zukunft bietet die Kombination aus BPEL4WS und Web Services damit eine echte Alternative zu den klassischen EAI-Werkzeugen. Andererseits ist es auch denkbar, dass EAI-Werkzeuge um eine BPEL Engine erweitert werden.

Prinzipiell lassen sich mit Web Services und Business Process Modeling Languages Integrationsszenarien jeglicher Art realisieren, da sie alle oben genannten Integrationsmerkmale unterstützen.

## 5 Zusammenfassung und Ausblick

Dass Integration heute ein wichtiges Thema in fast allen Firmen ist, zeigen aktuelle Statistiken (vgl. [Born03]). Als Reaktion auf diesen hohen Bedarf nach Integrationslösungen haben Hersteller aus verschiedenen Bereichen ihre Werkzeuge um (standardisierte) Integrationsschnittstellen erweitert bzw. eigenständige Integrationswerkzeuge auf den Markt gebracht. Neben diesen reinen Integrationswerkzeugen haben sich in den letzten Jahren Basistechnologien für die Integration (z.B. CORBA, Web Services) etabliert. Die damit entstandene Vielfalt an Technologien und Werkzeugen stellt mit der Integration beauftragte Personen vor eine schwierige Aufgabe.

Um das geeignete Integrationswerkzeug bzw. die geeignete Integrationstechnologie auswählen zu können, muss das Integrationsszenario genauer untersucht werden, und es sollte eine Vision entwickelt werden, welche weiteren Integrationen in den nächsten Monaten und Jahren angestrebt werden. Auf der Basis dieser Vision kann dann die derzeit für diese Vision geeignete Integrationslösung ausgewählt werden.

Insgesamt lässt sich feststellen, dass die verschiedenen Integrationstechnologien immer weiter zusammen wachsen. So unterstützen heute bereits eine Reihe von EAI-Werkzeugen Web Services, und Workflow Management Systeme setzen auf standardisierte Business Process Modeling Languages. Solange der Prozess des Zusammenwachsens jedoch noch nicht abgeschlossen ist, steht der Integrationsverantwortliche vor der Qual der Wahl.

### Literatur:

- [Bern96] Bernstein, Philip A. "Middleware: A Model for Distributed System Services," Communications of the ACM, February 1996, Vol. 39, No.2, 86-98.
- [BFGH02] Bunjes, B., Friebe, J., Götze, R. und Harren, A., Integration von Daten, Anwendungen und Prozessen am Beispiel des Telekommunikationsunternehmens EWE TEL, Wirtschaftsinformatik, 44, 2002, 5, 415-423
- [Born03] Born, A. und Diercks, J.: „Anwendungsintegration: grenzenlose Zusammenarbeit“, iX 11 (November 2003): 87-98
- [OMG02] Object Management Group: „Common Object Request Broker Architecture (CORBA/IIOP)“, [http://www.omg.org/technology/documents/corba\\_spec\\_catalog.htm](http://www.omg.org/technology/documents/corba_spec_catalog.htm)
- [BPML03] Business Process Modeling Language, <http://www.bpml.org>
- [BPEL03] BPEL4WS, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [Sun02] Sun Microsystems, Inc: „The Java Platform: Five Years in Review.“ <http://java.sun.com/features/2000/06/time-line.html>
- [MSOFT03] Microsoft, Inc: „Enterprise Solution Patterns Using Microsoft .NET“, <http://msdn.microsoft.com/practices/type/Patterns/Enterprise>
- [Mon02] Monson-Haefel, Richard: „Enterprise Java Beans“, O'Reilly, 2001
- [Nus00] R. Nussdorfer, Das EAI-Buch, E-Business und EAI, Integration von Anwendungen, Trends, Technologie und Lösungen, CSA Consulting, 2000



# Enterprise Application Integration als Enabler flexibler Unternehmensarchitekturen

Stephan Aier und Marten Schönherr

Technische Universität Berlin, Sekr. FR 6-7, Franklinstr. 28/29, 10587 Berlin,  
stephan.aier@tu-berlin.de, schoenherr@sysedv.tu-berlin.de,  
WWW home page: <http://www.sysedv.tu-berlin.de/eai>

**Zusammenfassung.** Ein wesentliches Ziel aktueller IT-Infrastruktur-Projekte ist die nachhaltige Flexibilisierung von Unternehmensarchitekturen. Der Beitrag erörtert, beginnend mit dem Paradigma der Flexibilität von Unternehmensarchitekturen, Möglichkeiten der integrierten Modularisierung von Organisations- und IT-Strukturen im Rahmen eines integrierten Architekturmanagements.

## 1 Problemstellung – Notwendigkeit flexibler Unternehmensarchitekturen

Das System Unternehmung ist den vielschichtigen Veränderungen seiner Umwelt ausgesetzt. Die drei wesentlichen Umweltdimensionen sind *Komplexität*, *Dynamik* und *Abhängigkeit* [1]. Insbesondere die Dimensionen Komplexität und Dynamik [2] verstärkt durch aktuelle Entwicklungen wie Globalisierung und Technisierung [3], stellen Unternehmungen vor Entscheidungs- und Handlungsprobleme zu deren Lösung sie grundsätzlich zwei Möglichkeiten der Reaktion haben [4]:

1. Sie können Maßnahmen zur Einwirkung auf die Umwelt mit dem Ziel ergreifen, Dynamik und Komplexität durch Verringerung der Abhängigkeiten zu reduzieren.
2. Oder sie können die Anpassungsfähigkeit der Unternehmung erhöhen.

Im vorliegenden Beitrag wird der zweite Punkt – die Erhöhung der Anpassungsfähigkeit – weiter untersucht. Dazu werden nach der Charakterisierung von Flexibilität als Gestaltungsziel die Architekturkomponenten Organisation und IT zueinander in Bezug gesetzt, um dann einen Ansatz zur Modularisierung der Organisations- und IT-Architektur im Rahmen von Enterprise Application Integration (EAI) darzulegen.

## 2 Flexibilität als Gestaltungsziel

Die Erkenntnis, dass „Unternehmenswandel von einem in größeren zeitlichen Abständen zu organisierenden Ereignis zu einem Dauerzustand geworden ist bzw. werden muss“ [5], ist heute weithin akzeptiert. Die Aufgabe besteht nun darin, Unternehmungen für diesen Wandel zu flexibilisieren.

KRÜGER differenziert den Wandel von Unternehmungen nach Wandlungsbedarf, Wandlungsbereitschaft und Wandlungsfähigkeit. [5] Die *Wandlungsfähigkeit* stellt den Kern der Betrachtungen dieses Beitrag dar.

Um mit intern und extern ausgelösten Wandlungsbedarfen effektiv umzugehen, sind in den Unternehmungen entsprechende Strukturen und organisatorische Instanzen zur Institutionalisierung von Wandel zu schaffen.<sup>1</sup>

Neben institutionellen Maßnahmen werden im Folgenden strukturelle Maßnahmen zur Erhöhung der Wandlungsfähigkeit – Maßnahmen zur Flexibilisierung – diskutiert. Der Begriff der *Flexibilität* entstammt dem Lateinischen und bedeutet Veränderbarkeit, Beweglichkeit oder Biogsamkeit. Ein System ist dann flexibel, wenn einem Wandlungsbedarf ein in angemessener Zeit aktivierbares Wandlungspotenzial im System gegenübersteht [4,9]. Der Wandlungsbedarf enthält eine sachliche und eine zeitliche Dimension. Die zeitliche Dimension fordert zum einen das Reaktionsvermögen einer Unternehmung als auch die Fähigkeit antizipativer Anpassung. HILL/FEHLBAUM/ULRICH bezeichnen dies als Produktivität zweiter Ordnung [10]. KIESER/KUBICEK identifizieren folgende Ausprägungen der Struktur von Unternehmungen als geeignet, deren Flexibilität zu erhöhen [4]:

- geringe Spezialisierung auf Stellen und Abteilungsebene
- starke Dezentralisierung
- flache Hierarchien
- Minimierung der Stärke zentraler unterstützender Abteilungen (Stäbe)
- einfache, d. h. keine umfassenderen Matrixstrukturen

Die genannten Ausprägungen zielen auf eine Entkopplung der Strukturen und Prozesse durch eine Reduktion von Schnittstellen. Im Folgenden werden diese Gedanken weiterentwickelt, um durch Modularisierung auf verschiedenen Ebenen der Unternehmung, Entkopplung und flexible Rekonfiguration von Architekturen zu ermöglichen.

### 3 Organisation und IT

Heute ist es nicht ausreichend ausschließlich formale organisatorische Gestaltungsaspekte bei der Flexibilisierung von Unternehmungen zu betrachten. Vielmehr ist es notwendig auch die informations- und kommunikationstechnologischen Aspekte zu berücksichtigen und beide in einem integrierten Architektur-Ansatz zu verbinden. Umgekehrt macht es wenig Sinn, Informationssysteme einzuführen oder zu ändern, ohne die Wechselwirkungen mit den Aufgaben und Prozessen die sie unterstützen zu berücksichtigen [11,12]. EAI integriert hier nicht nur IT-Systeme sondern ist vor allem der Auslöser, die Domänen Organisation und IT integriert zu gestalten und gemeinsam weiterzuentwickeln.

In der Wissenschaft hat diese Diskussion um die gegenseitigen Abhängigkeiten von IT und Organisation der Unternehmung eine lange Tradition, in der

<sup>1</sup> Bspw. kontinuierliche Prozessverbesserungen [6] durch ein betriebliches Vorschlagswesen [7] oder Kaizen [8].

sowohl die Technologie (*technological imperative*), die Organisation (*organizational imperative*) als auch komplexe Wechselwirkungen zwischen beiden (*emergent perspective*) [13] als treibende Faktoren beschrieben wurden [14,15,16,17].<sup>2</sup> FRESSE folgend, wird hier nicht davon ausgegangen, dass Informationstechnologie und Unternehmensorganisation in einem deterministischen Ursache-Wirkungs-Zusammenhang stehen [3]. Vielmehr wird angenommen, dass IT eine den Gestaltungsspielraum des Organisators erweiternde Option darstellt [19].

## 4 Modulare Unternehmensarchitekturen

Unter einer Unternehmensarchitektur wird hier das Zusammenwirken technologischer, organisatorischer und psychosozialer Aspekte bei der Entwicklung und Nutzung von betrieblichen soziotechnischen Informationssystemen verstanden [20]. Im Folgenden werden vor allem die technologischen und organisatorischen Domänen als IT- und Organisationsarchitekturen betrachtet. Ein aktuell diskutiertes Mittel zur Flexibilisierung von IT- und Organisationsarchitekturen ist die Modularisierung. Unter Zuhilfenahme der Systemperspektive werden nachfolgend allgemeine Charakteristika von Modulen beschrieben, welche der Organisations- und IT-Literatur entnommen sind und hier für beide Bereiche gleichrangig gelten sollen.

Ein *Modul* besteht aus zwei Teilen, der *Modulschnittstelle* und dem *Modulrumpf*. Die Modulschnittstelle enthält dabei die Spezifikation der Leistungen des Moduls die für seine Umwelt zur „Benutzung“ notwendig sind. Der Modulrumpf implementiert die spezifizierten Leistungen. *Modularisierung* bedeutet die Strukturierung eines Systems in kleine, teilautonome Subsysteme. Die Komplexitätsreduktion ergibt sich dabei aus der Subsystembildung innerhalb des Systems Unternehmung [21]. Die Subsystembildung wirkt komplexitätsreduzierend, da sie zum einen die subsysteminterne Komplexität im Sinne der Kapselung vor der Subsystemumwelt verbirgt, zum anderen durch die Reduktion auf wenige bekannte Schnittstellen eine Entkopplung der Subsysteme bewirkt.

Die Flexibilisierung der Architektur ergibt sich durch die nun leichtere Rekonfigurationsmöglichkeit der entkoppelten Module. Komplexitätsreduktion kann somit als Voraussetzung für die Flexibilisierung angesehen werden. Bei der Modularisierung sind folgende Gestaltungsziele zu beachten [22,23]:

- Abstraktion von der Implementierung
- Kapselung im Sinne des Verbergens der internen Funktionsweise
- Austauschbarkeit
- Wiederverwendbarkeit
- Zeitliche Gültigkeit
- Orthogonalität (im Sinne von „sich gegenseitig nicht beeinflussend“)
- Überschneidungsfreiheit
- Vollständigkeit (Abgeschlossenheit)

---

<sup>2</sup> Eine Übersicht über Studien sowie die Analyse von Ergebnissen finden sich in [18].

- Wohldefinierte Schnittstellen
- Schnittstellenminimalität<sup>3</sup>
- Generizität

Durch die Anwendung dieser Prinzipien werden Module geschaffen, die potenziell kombinierbar, wiederverwendbar und gut änderbar sind [22]. Im Folgenden wird nun dargestellt, welche Ansätze sich bei der Modularisierung der Organisation und der IT ergeben.

#### 4.1 Modularisierung der Organisation

Die Modularisierung der Organisation, wird im Folgenden wird auf der Ebene der Gesamtunternehmung als Makroebene und der Ebene der Geschäftsprozesse als Mikroebene betrachtet. Mit der Definition einer Makro- und einer Mikroebene werden zwei Ziele verfolgt:

1. Festigung und zugleich Flexibilisierung der Strukturen
2. Definition überschaubarer und managebarer Module

Die Makroebene hat eine stark strukturierende Wirkung auf die Organisation. Die Module der Makroebene, die sogenannten Makromodule, sollen über einen längeren Zeitraum stabil sein. Die Makroebene bildet damit ein Bezugssystem als Grundlage für die Gestaltung der Module der Mikroebene innerhalb der jeweiligen Makromodule. Die Module der Mikroebene hingegen sollen die Dynamik der wechselnden Prozessanforderungen durch entsprechende Rekonfiguration widerspiegeln. Ziel ist es also, eine bezuggebende, stabile, organisationsinvariante Ebene – die Makroebene – und eine dynamische, flexibel konfigurierbare Ebene – die Mikroebene – zu bilden. Nachfolgend werden die Charakteristika der Makro- und der Mikroebene dargestellt.

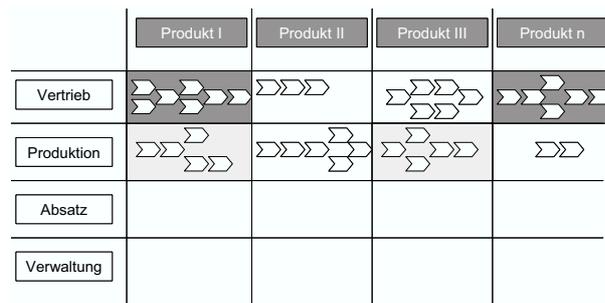
**Modularisierung auf Makroebene** Um eine direkte Abbildung strategieorientierter Organisationsstrukturen auf die IT-Infrastruktur zu erreichen, werden mehr als die klassischen ablauf- bzw. aufbauorganisatorischen Sichtweisen benötigt. Das Ziel der Definition der Makro-Module besteht in der Betrachtung der Gesamtorganisation mit Hilfe geeigneter Kriterien, um genau diesen Schritt ableiten zu können. Im Folgenden werden in Erweiterung der o. g. allgemeinen Kriterien spezielle Kriterien zur Bildung von Makromodulen genannt:

- Ähnlichkeit der Prozesse innerhalb eines Moduls
- Ähnlichkeit des notwendigen Prozess-Know-hows
- Minimierung der externen Abhängigkeiten
- Maximierung der internen Kohärenz
- Unabhängigkeit von operativen Prozessanpassungen

<sup>3</sup> In Bezug zur Systemtheorie bedeutet Schnittstellenminimalität die Reduktion der Relationen des Subsystems zu seiner Umwelt und somit Maximierung – im Sinne von Verschiebung – der Relationen in den Subsystemen.

- Branchenspezifische Best Practices für die Abgrenzung der Module

Die aufgezählten Kriterien werden z. T. im Bereich der Componentware verwendet, um die Granularität von Softwarekomponenten zu bestimmen. Die analoge Verwendung auf Makroebene führt zumindest zu ähnlichen Methoden, mit denen später ein Match zwischen Makro-, Mikro- und Softwarearchitekturebene erreicht werden kann. Die definierten Makro-Module stellen keine Handlungsanweisung für eine Reorganisation dar. Sie verfolgen vielmehr das Ziel, eine besondere Sichtweise auf die Organisation zu gewähren. Diese Sichtweise unterstützt die Modularisierung der jeweiligen IT-Systeme, die die Prozesse eines Makro-Moduls unterstützen. In Abbildung 1 wird anhand einer klassischen Matrixorganisation die sich nach Sparten und Funktionsbereichen aufstellt gezeigt, wie Makro-Module durch die oben aufgeführten Kriterien gebildet werden. Dabei stellen die farbig gleich hervorgehobenen Bereiche jeweils ein Makro-Modul dar. Beispielsweise wurden innerhalb der verschiedenen Vertriebsorganisationen zwei Bereiche als Makro-Modul definiert. Grund dafür könnte sein, dass sich die Prozesse in beiden Bereichen stark ähneln und ähnlicher Informationen zur Ausführung bedürfen.



**Abb. 1.** Definition von Makro-Modulen in einer Matrixorganisation

Auf der Ebene der Makromodule sollen Strukturähnlichkeiten die Grundlage für die Definition bilden. Es können situativ weitere Kriterien zu den oben genannten bestimmt werden. Prozesse laufen i. d. R. modulübergreifend, d. h. es wird keine Abgrenzung nach Anfang bzw. Ende von Prozessketten vorgenommen.<sup>4</sup>

**Modularisierung auf Mikroebene** Innerhalb der definierten Makromodule sollen jetzt die Strukturähnlichkeiten genauer betrachtet werden. Für die IT-Systemwelt sind vor allem die innerhalb der Module vorhandenen Geschäftsprozesse relevant. Hier müssen alle Prozessschritte verglichen und die

<sup>4</sup> Vgl. zu praktischen Umsetzungen ähnlicher Konzepte bei der Credit Suisse [24], bei der HypoVereinsbank [25].

strukturgleichen identifiziert werden. Ein Geschäftsprozess setzt sich prinzipiell aus verschiedenen Schritten zusammen. Die Wahrscheinlichkeit, dass sich in einer Menge ähnlicher Prozessketten identische Schritte befinden, ist sehr hoch. Selbst wenn sich die Teilprozesse nur sehr leicht unterscheiden, kann betrachtet werden, ob es im Sinne einer Komplexitätsreduktion möglich ist, die tatsächlichen Geschäftsprozesse so zu modifizieren, dass sie möglichst viele identische Teilprozesse verwenden.

GERYBADZE leitet aus den Methoden der modularen Produktgestaltung Kriterien für die Bestimmung des Grades der Modularisierbarkeit von Prozessen ab. Eine gute Modularisierbarkeit ergibt sich nach GERYBADZE, wenn [26]:

- sich für jede Aktivität eine genau definierte Funktion des Gesamtsystems definieren lässt,
- die Qualität des Ergebnisses dieser Aktivität genau bestimmt werden kann,
- für den jeweiligen Output Preise bzw. Verrechnungspreise bestimmt werden können und
- die Schnittstellen sehr genau definiert werden können.

Neben der generellen Frage der Modularisierbarkeit ist die Bestimmung der *optimalen Modulgröße* entscheidend für die erfolgreiche Umsetzung der Modularisierung auf Prozessebene. Zu große Module können die erhoffte Flexibilität der Organisation verhindern, da die Modularisierung ja gerade kleine, flexibel integrierbare Fragmente schaffen soll, die beweglicher sind als eine gesamte monolithische Struktur. Zu große Module können durch ihre Komplexität die Beherrschbarkeit der Prozesse innerhalb des Moduls einschränken. Andererseits bedeuten zu viele kleine Module eine stärkere Arbeitsteilung und Zergliederung. Je kleiner die Module, desto spezialisierter müssen naturgemäß die Prozesse innerhalb eines Moduls sein. Um allgemeine Kriterien zur Schaffung einer optimalen Modulgröße zu definieren, müssen die Charakteristika der Modularisierung betrachtet werden. Die zu beachtende *Mindestgröße eines Moduls* ergibt sich aus den Aktivitäten für ein klar definierbares (Zwischen-)Produkt. Die *maximale Modulgröße* wird durch die Beherrschbarkeit der Komplexität innerhalb eines Moduls bestimmt. Die Komplexität darf nicht so groß sein, dass die Verantwortlichen das Modul nicht mehr steuern können.

**Realisierungsprinzipien** Die Realisierung einer modularen Organisation ist stark von den situativen Rahmenbedingungen abhängig. Wesentliche Einflussfaktoren sind die *Größe* der Organisation, ihr *Angebotsprogramm*, ihre *Internationalisierung* und *Kultur*, die bestehende *Strukturierung* sowie die *Organisationsumwelt* [4]. Trotz dieser situativen Abhängigkeit lassen sich einige grundsätzliche Realisierungsprinzipien finden:

Das Hauptprinzip ist, *kleine Organisationseinheiten* zu bilden. Sie müssen groß genug sein, zusammengehörige Prozesse zu einem Objekt (z. B. Produkt, Produktgruppe etc.) zu umfassen. Sie dürfen aber den Umfang und die Komplexität betreffend die Aufnahmegrenzen und Problemlösungskapazitäten des Menschen nicht überschreiten.

Die *Prozessorientierung* kommt durch die Forderung zum Ausdruck, die Module durchgängig an den Prozessen zur Erstellung von Leistungen zu orientieren. Die Prozessorientierung zeigt, dass die Modularisierung nicht auf eine funktionsorientierte, sondern auf eine objektorientierte Strukturierung abzielt.

In engem Zusammenhang mit der Prozessorientierung steht die *Kunden-* bzw. *Marktorientierung*. Hiernach sollen sich alle Wertschöpfungsaktivitäten der Module an den externen und/oder internen Kundenanforderungen orientieren. Dies resultiert aus dem Hauptziel, der flexiblen Anpassung an die Erfordernisse des Marktes.

*Integriertheit der Aufgaben:* Die Prozesse in einem Modul sollen weitestgehend ihrer Art nach zusammengehören, um die Abgeschlossenheit der in einem Modul konzentrierten Prozesse zu gewährleisten.

*Nicht-hierarchische Koordinationsformen:* Die Koordination autonomer Handlungseinheiten, die nicht mehr in einem hierarchischem Verhältnis zueinander stehen, erfordert neue Formen der Zusammenarbeit, welche nicht mehr auf Fremdsteuerung, sondern auf Selbststeuerung basieren.

## 4.2 Modularisierung der IT

Ausgehend von der dargelegten modularen Organisationsarchitektur sind adäquate IT-Architekturen zu definieren. Die abgestimmte Gestaltung von Strategien, Prozessen und technischen Infrastrukturen ist ein klassisches Thema der Wirtschaftsinformatik [27]. Dazu wurden verschiedene Architekturkonzepte entwickelt, die eine Homologie zwischen Organisation und IT zum Ziel haben [28,29,30]. Diese Konzepte beruhen jedoch auf der Annahme, über Implementierung und Neueinführung von IT zu sprechen. Um möglichst nachhaltige Architekturen zu schaffen, sollte das Paradigma der *strukturellen Analogie* zwischen Organisation und IT so weit wie möglich aufrecht erhalten werden [20].

**EAI vs. Service Oriented Architecture (SOA)** Serviceorientierung und Systemintegration wurden viele Jahre unabhängig voneinander diskutiert. EAI-Plattformen integrieren heterogene Systemlandschaften zentral auf Prozess-, Methoden- und Datenebene.<sup>5</sup> Je stärker die Integrationsprojekte allerdings mit objektorientierten Verfahren implementiert werden, desto größer erscheint die Nähe zur Serviceorientierung. Die dienstorientierte Anwendungsintegration ist mit der Integration auf Interface- und Methodenlevel vergleichbar. Sie stellt eine alternative Sichtweise dar, die versucht, „gewrappte“ Module aus Altanwendungen und bereits serviceorientiert implementierte Neuanwendungen innerhalb einer SOA zu integrieren. Eines der wichtigsten Ziele der Serviceorientierung ist die Wiederverwendung bestehender Komponenten. Voraussetzung dafür ist ein zentrales Servicemanagement, das Funktionen wie Service-Life-Cycle, Serviceverteilung und die Versionierung bereitstellen muss [32].

<sup>5</sup> Vgl. zu den Begriffen User Interface Integration, Data Level Integration, Application Interface Integration, Method Integration, Service based Integration sowie Process Integration [31].

Da die meisten Entwicklungswerkzeuge inzwischen eine Serviceorientierung unterstützen, werden Anwendungen in Zukunft sehr viel einfacher als Services entwickelbar sein. Altanwendungen sind jedoch oft schwer in als Service abgrenzbare Module aufzuspalten. Neben Entwicklungsumgebungen die eine Neuimplementierung mit Servicecharakter aktiv unterstützen bieten einige Integrations-Suiten inzwischen ebenfalls Tools, die das Wrappen von Altanwendungen in Services erleichtern. Die Kosten dieser initialen Umwandlung können hoch sein und oft widerspricht der Eingriff in den Quellcode den strategischen Richtlinien der IT-Verantwortlichen, die vor allem monolithische Legacies unberührt lassen wollen [33]. Bei der Betrachtung einer Gesamt-IT-Architektur muss von zu integrierenden Altanwendungen und geplanten Neuimplementierungen ausgegangen werden. Sowohl der zentrale EAI-Ansatz als auch die dezentrale SOA stellen Methoden zu Verfügung, Lösungen für das beschriebene Spannungsfeld zu implementieren. EAI und SOA sind daher harmonisierende Bestandteile einer Gesamt-IT-Architektur.

**Problemfelder der technischen Modularisierung** Im Kontext der Systemintegration muss allerdings beachtet werden, dass eines der primären Ziele der EAI, die Reduktion der Punkt-zu-Punkt-Integrationsszenarien nicht durch eine ähnlich komplexe SOA ersetzt wird. Zwei Hauptprobleme bei der Anbindung von Altanwendungen sind standardisierte Schnittstellen und die Granularität der Services. Bei der serviceorientierten Integration bestehender Anwendungen kommt es auf die Granularität der als Services verpackten Funktionen an. So macht es Sinn, diejenigen Funktionen als Services zugänglich zu machen, die allgemein benötigt und somit wieder verwendet werden sollten. Sie sollten eine komplette Arbeitseinheit ausführen sowie in ihrer Funktion und im Ergebnis gut beschreibbar sein [34]. Mit größter Wahrscheinlichkeit liegen die Altanwendungen jedoch monolithisch bzw. in falscher Granularität vor [35]. Das Problem der Granularität stellt sich auch für Anwendungen, die in Programmiersprachen geschrieben wurden, die mehr oder weniger direkt unter Anwendung spezieller Tools in Webservices umgewandelt werden können. Obwohl sich die Verzeichnisse, die Services beschreiben aus Definitionen von Schnittstellen bereits bestehender Technologien verteilter Softwaresysteme (Java-Klassen, JavaBeans, CORBA-Objekte, Visual Basic-Klassen, C#) erzeugen lassen, sind diese zumeist ebenfalls nicht in der für Services geeigneten Granularität definiert [36].

## 5 Fazit

Viele Publikationen mit technischem Fokus diskutieren derzeit dezentrale Architekturen zur Integration komplexer IT-Infrastrukturen. Unternehmen greifen das Thema der Wiederverwendung erneut auf, obwohl es vor einigen Jahren unter dem Namen der Business Process Repositories, die konzipiert wurden, um rekonfigurierbare Geschäftsprozesse zu implementieren, fast völlig aus der Fachdiskussion verschwunden waren. Performantere Plattformen begründen teilweise diese unerwartete Renaissance. Das Thema EAI trägt ebenfalls zur aktuellen

Diskussion bei, die auf Unternehmensarchitekturebene geführt wird. Dabei geht es im ersten Schritt um die fachliche Definition der Module (bzw. Services), nachfolgend werden erst Technologien zur Implementierung eine Rolle spielen. In diesem Sinne konzentriert sich der Beitrag vor allem auf die Modularisierung auf fachlicher Ebene und deren Überführung in eine technische Ebene. Details der technischen Implementierung stehen nicht im Vordergrund.

## Literatur

1. Jurkovich, R.: A core typology of organizational environments. *Administrative Science Quarterly* **19** (1974) S. 380–394
2. Krystek, U.: Vertrauen als Basis erfolgreicher strategischer Unternehmensführung. In: D. Hahn/B. Taylor, *Strategische Unternehmensplanung – Strategische Unternehmensführung*. 8 edn. Physica, Heidelberg (1999) S. 266–288
3. Frese, E.: *Grundlagen der Organisation: Konzept – Prinzipien – Strukturen*. 8 edn. Gabler, Wiesbaden (2000)
4. Kieser, A., Kubicek, H.: *Organisation*. 3 edn. De Gruyter, Berlin, New York (1992)
5. Krüger, W.: Management permanenten Wandels. In: H. Glaser, E.F. Schröder, A. v. Werder, eds. *Organisation im Wandel der Märkte*. Gabler, Wiesbaden (1998) S. 227–249
6. Österle, H.: *Business Engineering: Prozeß- und Systementwicklung*. 2 edn. Volume 1. Springer, Berlin et al. (1995)
7. Thom, N.: *Betriebliches Vorschlagswesen: ein Instrument der Betriebsführung und des Verbesserungsmanagements*. 5 edn. Lang, Berlin et al. (1996)
8. Imai, M.: *Kaizen*. Ullstein, Frankfurt a. M., Berlin (1993)
9. Gronau, N.: Modellierung von Flexibilität in Architekturen industrieller Informationssysteme. In: H. Schmidt, *Modellierung betrieblicher Informationssysteme. Proceedings der MobIS-Fachtagung, Siegen (2000)* S. 125–145
10. Hill, W., Fehlbaum, R., Ulrich, P.: *Organisationslehre 1: Ziele, Instrumente und Bedingungen der Organisation sozialer Systeme*. 5 edn. Haupt, Bern, Stuttgart, Wien (1994)
11. Kaib, M.: *Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele*. DUV, Wiesbaden (2002)
12. Derszteler, G.: *Prozessmanagement auf Basis von Workflow-Systemen*. Josef Eul, Lohmar, Köln (2000)
13. Markus, M., Robey, D.: Information technology and organizational change: Causal structure in theory and research. *Management Science* **34** (1988) S. 583–589
14. Leavitt, H., Whisler, T.: Management in the 1980s: New information flows cut new organization flows. *Harvard Business Review* **36** (1958) S. 41–48
15. Applegate, L.M., Cash, J.I., Miles, D.Q.: Information technology and tomorrow's manager. *Harvard Business Review* **66** (1988) S. 128–136
16. Rockart, J.F., Short, J.E.: It in the 90's: Managing organizational independence. *Sloan Management Review* (1989) S. 7–17
17. Burgfeld, B.: *Organisationstheorie und Informationstechnologie*. DUV, Wiesbaden (1998)
18. Lewin, A.Y., Hunter, S.D.: Information Technology & Organizational Design: A Longitudinal Study of Information Technology Implementations in the U.S. Retailing Industrie, 1980-1996. In: H. Glaser, E.F. Schröder, A. v. Werder, eds. *Organisation im Wandel der Märkte*. Gabler, Wiesbaden (1998) S. 251–286

19. Frese, E.: Theorie der Organisationsgestaltung und netzbasierte Kommunikationseffekte. In: E. Frese, H. Stöber, eds. E-Organisation. Gabler, Wiesbaden (2002) S. 191–241
20. Gronau, N.: Wandlungsfähige Informationssystemarchitekturen – Nachhaltigkeit bei organisatorischem Wandel. Gito, Berlin (2003)
21. Krcal, H.C.: Systemtheoretischer Metaansatz für den Umgang mit Komplexität und Nachhaltigkeit. In: R. Leisten and H.-C. Krcal, Nachhaltige Unternehmensführung – Systemperspektiven. Gabler, Wiesbaden (2003) S. 3–30
22. Rombach, D.: Software nach dem baukastenprinzip. Fraunhofer Magazin (2003) S. 30–31
23. Lang, K.: Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen. Gabler, Wiesbaden (1997)
24. Hagen, C.: Integrationsarchitektur der Credit Suisse. In: S. Aier, M. Schönherr, eds., Enterprise Application Integration – Flexibilisierung komplexer Unternehmensarchitekturen. Gito, Berlin (2003) S. 61–83
25. Weber, H.-W.: Wege zur Entwicklung von Wertschöpfungsnetzen: Konzeption einer modularen IT-Architektur. <http://www.sysedv.tu-berlin.de/eai/eaitag200302> Vortrag auf dem EAI-Expertentag, Berlin (2003)
26. Gerybadze, A.: Strategisches Management und dynamische Konfiguration der Unternehmens-Umwelt-Beziehungen. In: R. Leisten and H.-C. Krcal, Nachhaltige Unternehmensführung – Systemperspektiven. Gabler, Wiesbaden (2003) S. 83–100
27. Wall, F.: Organisation und betriebliche Informationssysteme – Elemente einer Konstruktionslehre. Gabler, Wiesbaden (1996)
28. Krcmar, H.: Bedeutung und ziele von informationssystem-architekturen. Wirtschaftsinformatik **32** (1990) S. 395–402
29. Pohland, S.: Globale Unternehmensarchitekturen – Methode zur Verteilung von Informationssystemen. Weißensee-Verlag, Berlin (2000)
30. Scheer, A.W.: Architektur integrierter Informationssysteme. Springer, Berlin et al. (1991)
31. Aier, S., Schönherr, M.: Flexibilisierung von Organisations- und IT-Architekturen durch EAI. In: S. Aier, M. Schönherr, eds., Enterprise Application Integration – Flexibilisierung komplexer Unternehmensarchitekturen. Gito, Berlin (2003) S. 1–59
32. Lublinsky, B., Farrell, M.: 10 misconceptions about web services. EAI Journal (2003) S. 30–33
33. Apicella, M.: Side by side in perfect harmony? InfoWorld (2002)
34. Narsu, U., Murphy, P.: Web services adoption outlook improves. (2003) Giga Information Group.
35. Erlikh, L.: Integrating legacy into extended enterprise: Using web services. (2003) Relativity Technologies.
36. Newcomer, E.: Understanding Web services: XML, WSDL, SOAP and UDDI. Addison-Wesley Longman, Amsterdam (2002)

# Integrating product catalogs via multi-language ontologies

Manfred A. Jeusfeld<sup>1</sup>

Tilburg University, CRISM/Infolab, Postbus 90153, NL-5000 LE Tilburg,  
manfred.jeusfeld@uvt.nl

**Abstract.** A vertically integrated market consists of potentially competing companies who supply each other with products and services. Information technology can be employed to improve the business transactions and the flow of information within the market. In this paper, we show a solution on integrating massively heterogeneous product catalogs into multi-language and multi-role product groups organized as ontologies. Besides product classification, the method also addresses the classification of product properties. The product catalog data structure is decomposed into a set of label-value pairs which are then classified into the multiple ontologies which is the basis for query formulation. Our approach has been realised within the European research project MEMO (Mediating and Monitoring Electronic Commerce) using the construction industry in an extensive case study. The method can also be applied to related areas like multi-media databases or digital libraries.

## 1 Introduction

Enterprise application integration requires to understand the heterogeneity of services and data required for coupling the systems. Federated databases are one technology to achieve that but it focusses on data structure integration rather than semantic integration.

In this paper, we present an approach to link the content of databases on a fine grain to an ontology that allows to look up the data and even to formulate queries on top of the ontology. Essentially, the content of the database is viewed as a collection of label-value pairs rather than tuples. These label-value pairs are reified, i.e. regarded as identifiable objects, and then classified into the ontology. The ontology is featuring attribute concepts to which the label-value pairs are classified into.

The method requires the databases to be of a certain format: all attributes in a relations are functionally dependent on a single key attribute. Product catalogs are a prominent example of such relations. The information provider has to take into account that potential information consumers have different roles in the enterprise. Moreover, there might be different catalogs exporting information about the same product. To summarize, the obstacles to be addressed in this paper are:

- The structure problem: an information provider structures the product details in proprietary format. How can a customer match her query against multiple product catalog structures?

- The semantics problem: a natural language term can be ambiguous. How can a query be formulated such that there is no ambiguity in the interpretation of a search term?

There are other obstacles like the unique identification of products. This is not covered by our paper. We assume that products are identified using a unique coding scheme like the EAN code. If that is not available, techniques from data warehousing have to be employed to generate unique object identifiers.

## 2 Classifying product details

Our solution to bridge the structural and semantic gap between the publishers of product catalogs and the customers is organized in three steps. First, to model the key terms of each customer group are represented in so-called ontologies. Second, the product catalog structure is related to the ontologies by classification links. Finally, a query mechanism is deployed to extract answers from the product catalogs based on queries expressed in terms of an ontology.

We use the representation framework of Telos [Mylopoulos et al., 1990] as implemented in ConceptBase [Jarke et al., 1995] to uniformly represent all meta models, schemas, and data.

### 2.1 A multilingual ontology model

The ontology meta model in figure 1 features the central entity 'Concept'. A concept can be related to other concepts, e.g. the 'sbk43\_2' (tile) concept is related to the concept 'sbk43' (floor covering). Any concept has translations into multiple languages. Concept attributes are special concepts which are about describing a property of an entity. For example, 'hcp23' (size) is a term to describe the physical dimension of an entity.

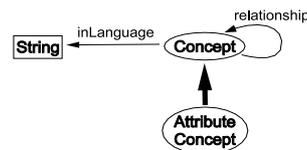


Fig. 1. Ontology meta model

Example ontologies are instantiated from the ontology meta model as shown in figure 2. The SBK ontology is an ontology for architects developed in the Scandinavian domain and being taken over by more and more European countries. It classifies products according to their function in a building. Formally, the term 'SBKConcept' is a subclass of 'Concept' since any SBK concept is also a concept. The HCP ontology is an attempt to classify product attributes. Hence, the class 'HCPConcept' is a subclass of 'AttributeConcept'.

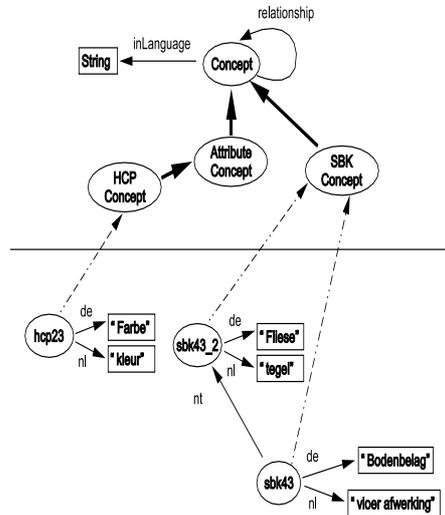


Fig. 2. Example ontologies

## 2.2 Linking product catalogs to ontologies

A product catalog is published as a relational table where the table structure is completely under the control of the product vendor <sup>1</sup>. The requirement for such a table data structure are:

- A product catalog contains one field which *identifies* the product about which a statement is made.
- A product catalog contains one or more fields which contain *product group* codes. A product group is subsuming products of the same kind, e.g. tiles.
- The remaining fields are *describing* some properties of the product.

Figure 3 displays the abstract model of product catalogs used in this paper. The bottom level is the raw data (i.e. product tuples). The name p123 is a tuple identifier. The middle level is the product catalog schema with TregaTiles being the relation name. The top level is the product catalog data model. The link between ProductProfile and Domain represents roughly the relational data model, the right hand side list our specializations, namely that the is a product identifier and there are fields product grouping. The publisher field will be used in section 3 to manifest ownership and traceability.

## 2.3 Classifying product properties

The challenge with product catalogs is not so much the classification of products into multiple product ontologies. Different areas like the chemical industry have developed

<sup>1</sup> Other formats like XML have been proposed but they should rather be seen as exchange formats. As product catalogs stem from the internal databases of an information provider, it is justified to start from a table structure.

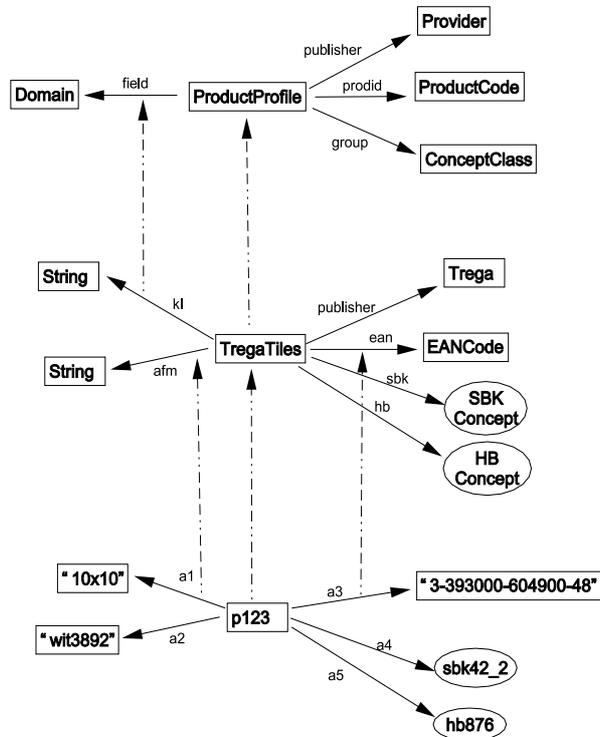


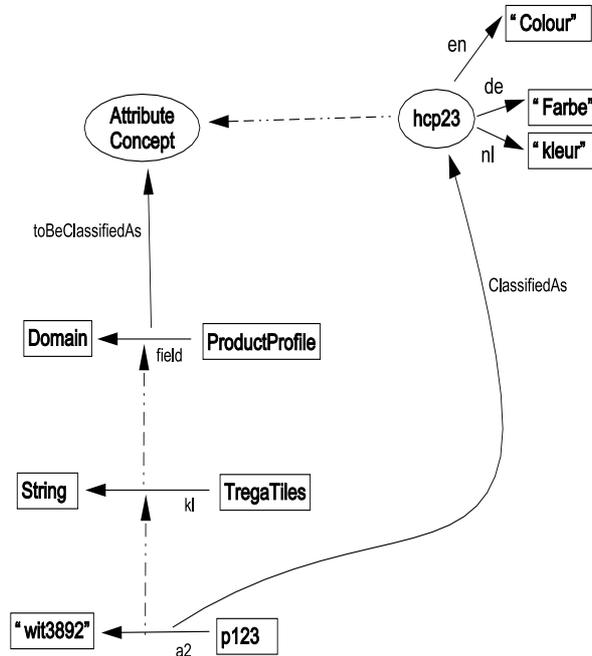
Fig. 3. Abstract model of product catalogs

hierarchical codes for product groups that any supplier of a product catalog can employ to classify her product offerings. A typical user [Callahan and Koenemann, 2000] limits the search space of a query by categories (product groups) and *then* proceeds with restrictions on the attributes. Hence, the challenge is to classify product profile fields like a1 and a2 into attribute concepts of ontologies.

Our restriction to product catalogs implies a certain structure of the database tables: each attribute is functionally dependent on key attribute (the product identifier). So essentially we restrict ourselves to databases that contains tables for entity types but not relationship types. This appears a strong restriction but the reader should keep in mind that we provide a new form of integrating such databases via a dedicated 'knowledge level' in the form of ontologies.

The classification into ontologies is realized in two steps. First, the products, identified by their keys, are classified into the concepts mentioned in the product group attributes of the original table. This step is straightforward and does not add any information. Second, the describing properties are mapped to so-called *attribute concepts* (see figure 4).

Attribute concepts are concepts occurring in some (natural) language to describe properties of things. Typical examples are size, color, and fire resistance. They share the



**Fig. 4.** Classifying product properties into attribute concepts

fact that some objects (like products) may have fillers for those properties. One might argue that the field names of the product catalogues already carry this information but

- field names in tables are often abbreviated and hard to read,
- some fields in some tables contain several values, in particular when the field type is a string type,
- some values are scattered to more than one fields, e.g. the number value and the unit code,
- subtype relationships between fields cannot be expressed, e.g. the length and the height a product are both specializations of the 'dimension' of the product,
- field names are expressed in some natural language, e.g. English, which makes access via equivalent terms in other languages difficult.

We assume that the describing catalog fields are classified into the attribute concepts by tuples

$$TOBECLASSIFIEDAS(R, j, AC)$$

where  $R$  is the name of the product catalog and  $j$  is the position of the describing attribute and  $AC$  is the attribute concept. Note that the combination  $(R, j)$  of the catalog name and the position of the attribute identifies the attribute at the schema level. Figure 4 denotes this graphically by the link *toBeClassifiedAs* attached at field attributes

of `ProductProfile`. `ConceptBase` uses expressions like `TregaTiles!kl` to denote the attribute `kl` of product catalog `TregaTiles`. For sake of readability, we will use a relational representation in this paper.

The attribute concepts in our ontology are identified and have links to their translations in as many natural languages as shall be supported by the system. What remains to be done is the classification of product catalog fields into attribute concepts. This is done by generating two types of so-called *universal catalog items*. Let  $R(ID, G_1, G_2, \dots, A_1, A_2, \dots)$  be a product catalog table with grouping attributes  $G_i$  and describing attributes  $A_j$ . Let further  $R(id, \dots, g_i, \dots, a_j, \dots)$  be some tuple of that catalog.

1. For each  $g_i$  generate a tuple  $C_1(id, R, i, g_i)$ . This realized the mapping of products to their product group: product  $id$  is classified into product group  $g_i$  due to the grouping attribute  $(R, i)$ .
2. For each  $a_j$  and each  $TOBECLASSIFIEDAS(R, j, AC)$  generate a tuple  $C_2(id, R, j, a_j, AC)$ . This classifies describing attributes at the tuple level into the attribute concepts of the ontology: the describing attribute  $(R, j)$  with value  $a_j$  is classified into attribute concept  $AC$ .

The second tuple  $C_2$  realizes the attribute classification. The method is loss-less in the sense that the original relation  $R$  can be reconstructed from the relations  $C_1$  and  $C_2$ . In figure 4, the second attribute `a2` of tuple number 123 is classified into attribute concept `hcp23`. The subsequent Datalog program patterns show that  $C_1$  and  $C_2$  can be constructed from  $R$  and vice versa:

```

C1 ( ID, R, i, Gi ) :- R ( ID, . . . , Gi, . . . ) .
C2 ( ID, R, j, AJ, AC ) :-
    TOBECLASSIFIEDAS ( R, j, AC ) ,
    R ( ID, . . . , AJ, . . . ) .

R ( ID, G1, . . . GM, A1, . . . , AN ) :-
    C1 ( ID, R, 1, G1 ) , . . . , C1 ( ID, R, M, GM ) ,
    C2 ( ID, R, 1, A1, _ ) , . . . , C2 ( ID, R, N, AN, _ ) .

```

Note that the same attribute  $(R, j)$  can be classified into several attribute concepts  $AC$ . We assume that each describing attribute  $(R, j)$  is classified into at least one such attribute concept.

The parameter `ID` is the product identifier (e.g. the EAN code of the product). The parameter `Gi` is the  $i$ -th grouping attribute of product catalog  $R$ , the parameter `AJ` is the  $j$ -th describing attribute of  $R$ . We include position numbers in the universal catalog items to ensure the *re-constructability* of the original relation. Each grouping attribute  $G_i$  of a product catalog  $R$  leads to the generation of one rule  $C1 ( ID, i, Gi )$ . Analogously, each describing attribute is matched by one rule  $C2 ( ID, R, j, AJ, AC )$ .

The ontology level adds additional query capabilities based on the new relations  $C_1$  and  $C_2$ . First, an attribute value can be retrieved via its attribute concept identified  $AC$ . Since this has links to multiple translations, the product can be queried using terms of any natural language. Second, attribute values hidden in a complex attribute (e.g. size and fireresistance on one text fields) can be retrieved by using the relevant attribute

concept that the user is interested in. Third, and most important, the query evaluator can exploit sub-typing relationships between attribute concepts formulated in the ontology. For example, queries on the 'dimension' of a product return both 'length' and 'height' properties. retrieved by the specific attribute concept

Figure 5 shows how data from multiple product catalogs are integrated. The classification method sketched above has to be executed on both catalogs and then allows to access the data using ontology terms rather than table field names. When the source tables come from different organization, the table name  $R$  can be include into the  $C_2$  table. Then, the answer can be traced back to the original source table.

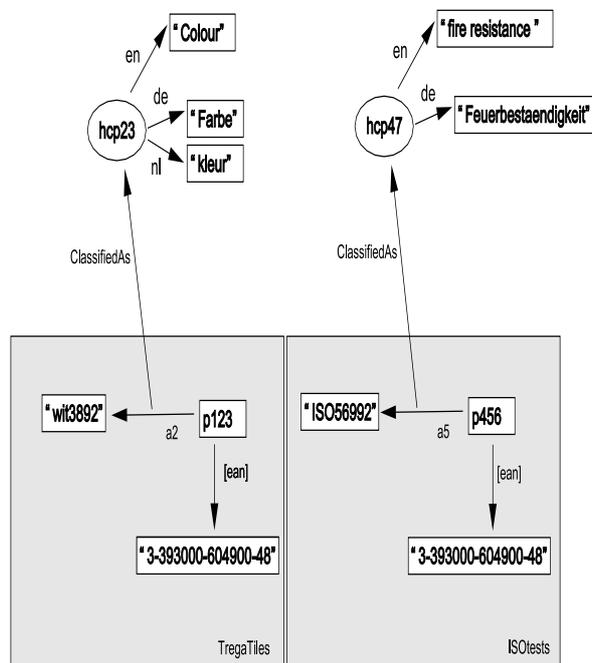


Fig. 5. Integrating product data from multiple catalogs

Figure 6 visualized the treatment of complex attribute values. Here, textual attribute is classified into two attribute concepts (for color and for fire resistance).

Complex attributes are likely in product catalogs from which human-readable representations (flyers, web pages) are generated. Our method makes their content accessible via a semantic layer.

## 2.4 Querying via the ontology

The Datalog rules show that the  $C_1$  and  $C_2$  relations do not remove information. But the question is: What is the benefit? First, the universal catalog representation allows

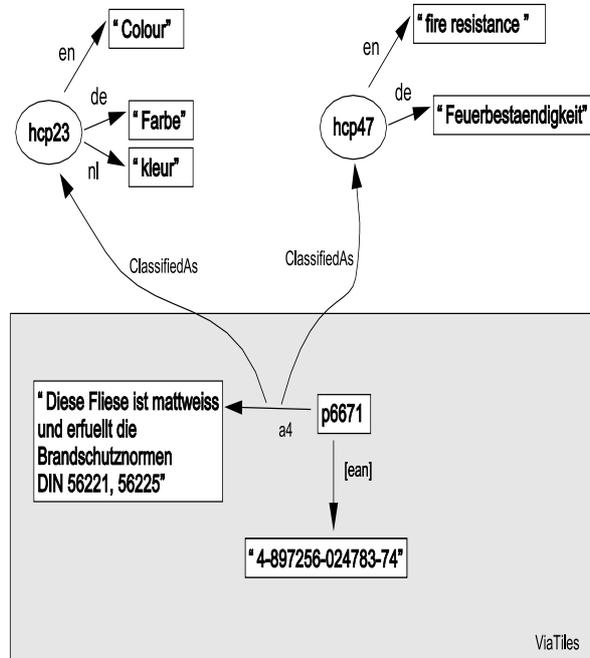


Fig. 6. Integrating product data from multiple catalogs

the integration of arbitrary many product catalogs. It is even possible that information about a given product is provided via multiple sources, e.g. the vendor's product catalog and the quality data supplied by some independent institute.

The more important benefit is however that the universal catalog items can be accessed via the ontology. This makes the user independent from the terms used by the supplier to represent the catalog schema. Moreover, the user can navigate in the network of relationships between ontology concepts in order to specialize or generalize the query. We discuss a few examples to show the added value of the ontology.

We assume a predicate  $P(G_1, rel, G_2)$  to denote that product group  $G_1$  is related to a product group  $G_2$  by an ontological relation  $rel$ . A possible value for  $rel$  is  $nt$  (narrower term). Further we assume a predicate  $T(N, L, A)$  which returns for a concept name  $N$  in language  $L$  its identifier  $A$  in the ontology.

Example 1: Show products of a given product group 'tile' and product groups narrower than 'tile'.

```

Q1(ID) :-
    T('tile', 'english', G),
    C1(ID, _, _, G).
Q1(ID) :-
    T('tile', 'english', G),

```

```

P(G,nt,G1),
C1(ID,_,_,G1).

```

The example can easily be extended to cover also broader product groups and to cover product groups that have a distance larger than 1 to the given group 'tile'. The Datalog rules are just for showing the feasibility of the query. A real user wouldn't code Datalog but just select the product group from the ontology.

Example 2: Show products of a given group 'tile' that have some information about fire resistance or a related attribute concept.

```

Q2(ID,V) :-
  T('fire resistance','english',F),
  REL(F,F1),
  T('tile','english',G),
  C1(ID,_,G),
  C2(ID,_,_,V,F1).
REL(F,F).
REL(F,F1) :- P(F,_,F1).

```

Like in the example before, the distance of concepts related to fire resistance can be larger than 1. Support of languages other than English is obvious.

A characteristic of our approach is that it makes no specific assumption about the value types of describing attributes in product catalogs. All values are treated as vanilla. This apparently makes the use of typed functions like comparison operators between numbers difficult if not impossible.

A possible way out is to represent a type lattice next to the attribute concepts in order to imply the value type of an attribute concept. We won't however. The more type information is represented the more typing conflicts can occur and potentially endanger the universality of our approach.

### 3 Traceability and Ownership of Data Items

The classification of multiple product catalogs implies the merge data items from these catalogs into one uniform space. This has the advantage of universal queries ranging over data of all catalogs. However, there are two concerns that need to be addressed: ownership and traceability.

1. Suppliers of product catalogs want to execute some control who can access their product data. Specifically, they want to be able to remove some or all of their data.
2. Users of the universal product catalog need to be able to trace back answers to the sources. Answers that cannot be traced back can also not be verified by the user.

Both requirements can be addressed by augmenting the universal catalog items  $C_1$  and  $C_2$  by a supplier identifier  $S$  and a tuple identifier  $T$ .

For a given product catalog  $R(ID, G_1, G_2, \dots, A_1, A_2, \dots)$  from some supplier  $S$  we create an ownership-aware variant  $OR(S, R, T, ID, G_1, G_2, \dots, A_1, A_2, \dots)$  which

just adds the constant values  $S$  and  $R$  and a tuple identifier which uniquely identifies the position of a tuple in the original relation  $R$ .

From this new base relation, we derive ownership-aware variants of the universal catalog items as follows:

```

OC1(S,T, ID,R, i, Gi) :- OR(S,R,T, ID, ..., Gi, ...).
OC2(S,T, ID,R, j, AJ, AC) :-
    TOBECLASSIFIEDAS(R, j, AC),
    OR(S,R,T, ID, ..., AJ, ...).

```

For each source product catalog  $OR$  and each attribute position  $i$  and  $j$  such rules  $OC1$  and  $OC2$  are generated. The augmented rules allow any universal catalog item be traced back precisely to the location where it originated from.

The supplier of some product catalog can easily compute the universal catalog items that are generated from its catalogs. Due to the logic-based definition, the effect of a change (items added, items removed) can incrementally be computed. This allows simple maintenance of the integrated product catalog from its sources.

Authorization rules based on ownership-aware variants can be employed to restrict access to certain catalog items. A simple authorization rule might be that only users of a user group  $gr_1$  are allowed to access information generated from the product catalog  $R$ :

```

OC1(S,R,T, ID, i, Gi) :-
    user(U), member(U, gr1),
    OR(S,R,T, ID, ..., Gi, ...).
OC2(S,R,T, ID, ID,R, j, AJ, AC) :-
    user(U), member(U, gr1),
    TOBECLASSIFIEDAS(R, j, AC),
    OR(S,R,T, ID, ..., AJ, ...).

```

The authorization can also be formulated in terms of product groups (proper concepts) and attribute concepts. This allows to target the information to user groups depending on the concepts they are interested in rather than the data structure as common with database systems.

## 4 Implementation Aspects

The approach has been realized using the ConceptBase system. ConceptBase has the ability to represent tuple, schema, and meta model information in the same framework based on Datalog as query language. Tuples are automatically decomposed into objects  $P(o, t, i, y)$  where  $t$  is the tuple identifier,  $i$  is the position of the attribute and  $y$  is the attribute value. The attribute itself is identified by  $o$ .

The import of product catalogs into the ConceptBase system is automatic. Just the location of the original database and the relation structure have to be specified. The classification of products in product groups is automated by the rules for  $C_1$ . The classification of describing attributes into attribute concepts requires the manual specification of the links `toBeClassifiedAs`. It has to be done once per relation.

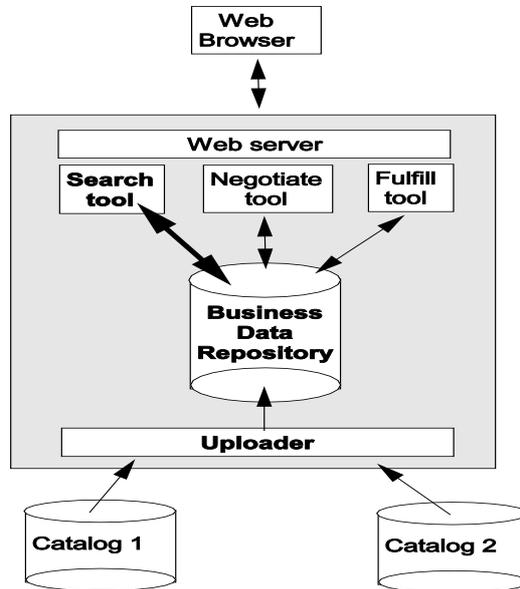


Fig. 7. Architecture of the MEMO system

Figure 7 displays the integration of product catalogs in the MEMO project. The relational uploader of ConceptBase integrates heterogeneous catalogs into the central repository. The repository contains also (multiple) ontologies into which the catalog items are classified into by the rules. The search tool is invoked by a web browser. A user query (given by a combination of ontology concepts) is translated into a Datalog rule and the result is returned in HTML format.

## 5 Conclusion

We presented a method to classify product details from heterogeneous product catalogs into multiple, multi-language ontologies in order to support the search for products using the search terms of professional user groups. Multi-linguality is fully supported on the level of ontologies and hence the level of query formulation. Entries in the product catalogs are however not available in multiple languages.

The main innovation lies in the introduction of *attribute concepts* and the classification of product profile fields into them. Attribute concepts make users independent from the somewhat arbitrary data structures chosen by catalog providers. The reader should note that attribute concepts do not make any assumption about the type of the attribute values. If an attribute is classified into an attribute concept, then we can just conclude that the attribute values is about the attribute concept. This relaxation on the data integration allows enormous flexibility in the structure of product catalogs that are supported by our approach. The only strong assumption is that each product profile should contain a uniform product identifier, in our case the EAN code.

The price to be paid is that queries like "all tiles with size greater than 10x10" cannot be answered directly. The business data repository simply makes no assumptions about the attribute value type of the size attribute.

The method for product profile classification has been implemented using the ConceptBase system. In the current prototype, product catalogs are copied into the business data repository. A possible extension is to keep the catalogs at the providers site and only maintain the ClassifiedAs relation in the central repository. The classification method was demonstrated with the example of product catalogs. It can be applied to any collection of catalogs where some entities are classified (products, companies, customers) and where these entities have descriptive attributes.

Future work shall investigate attribute typing to support the comparison of attribute values. While not all attributes are subject to comparison, it should be possible to maintain a type lattice aside the ontology into which attribute values are classified into. A type lattice that just mirrors the relational schema would allow for supporting the same type of queries that are possible on the original catalogs. The challenge is to select types for attribute concepts that are not a one-to-one counterpart of the relational attribute and to design type conversion routines to transform values between types. Basic techniques for this are already known. Our ontology-based approach just allows to decide to which degree inter-operability is desired. The vanilla type used in our current implementation is good enough to generate human-readable answers for product searches over multiple catalogs.

Another extension is to integrate process information onto the ontology. For example, the architect knows about an activity 'lay tiles' which includes tiles (sbk43.2) and some kind of glue. A process-aware query would be to find fire resistance information about all products involved in the 'lay tiles' activity.

**Acknowledgements.** This work was supported in part by ESPRIT project 26895 (MEMO: Mediating and monitoring electronic commerce). I would like to thank my colleagues esp. Kees Leune, Mareike Schoop and Christoph Quix for fruitful discussions. Special thanks go to Peter Raadsheer who helped us make the contact with the Dutch construction industry.

## References

- [Callahan and Koenemann, 2000] Callahan, E. and Koenemann, J. (2000). A comparative usability evaluation of user interfaces for online product catalog. In *Proceedings of the 2nd ACM conference on Electronic commerce, October 17 - 20, 2000, Minneapolis, USA*, pages 197–206. ACM.
- [Jarke et al., 1995] Jarke, M., Gallersdörfer, R., Jeusfeld, M., and Staudt, M. (1995). Concept-Base - a deductive object base for meta data management. *Journal of Intelligent Information Systems*, 4(2):167–192.
- [Mylopoulos et al., 1990] Mylopoulos, J., Borgida, A., Jarke, M., and Koubarakis, M. (1990). Telos: A Language for Representing Knowledge about Information Systems. *ACM Transactions on Information Systems*, 8(4):327–362.

# IT-basierte Integration in medizinischen Versorgungsnetzen

Mario Beyer<sup>1</sup>, Klaus A. Kuhn<sup>1</sup>, Christian Meiler<sup>2</sup>, Stefan Jablonski<sup>2</sup> und Richard Lenz<sup>1</sup>

<sup>1</sup> Philipps-Universität Marburg, Institut für Medizinische Informatik,  
Bunsenstr. 3, D-35033 Marburg.

{beyerm|kuhn|lenzr}@med.uni-marburg.de

<sup>2</sup> Friedrich-Alexander-Universität Erlangen, Lehrstuhl für Informatik 6,  
Martensstr. 3, D-91058 Erlangen.

{christian.meiler|stefan.jablonski}@informatik.uni-erlangen.de

**Zusammenfassung.** Die Versorgung von Patienten entwickelt sich zunehmend zu einem institutionsübergreifenden Behandlungsprozess. Daraus entsteht ein Bedarf für Informationssysteme, die geeignet sind, Prozesse im Versorgungsnetz angemessen zu unterstützen. Solche Systeme haben ein hohes Potential zur Verbesserung der Qualität. In dem Forschungsprojekt „Grundlagen prozessorientierter Informationssysteme im Krankenhaus und im Versorgungsnetz“ wurden die Anforderungen an solche Systeme untersucht. Dabei wurden Flexibilität, Erweiterbarkeit und Integration als zentrale Eigenschaften einer Systemarchitektur identifiziert. Dieser Artikel beschreibt funktionale Aspekte für eine Architektur und diskutiert die verschiedenen Aspekte bei der Integration von Daten heterogener Systeme. Ein Schwerpunkt liegt auf der Behandlung semantischer Inkompatibilität, wozu ein Abstraktionskonzept skizziert wird, das einen kaskadierten Ausbau der semantischen Kompatibilität verschiedener Teilnehmersysteme ermöglichen und damit falls nötig eine weitgehend ontologieunabhängige Verarbeitung von Dokumenten erlauben soll.

## 1 Einleitung

### 1.1 Grundlagen und Zielsetzung

Die fortschreitende Bildung von regionalen Gesundheitsnetzwerken, in denen sich Kliniken und Arztpraxen zusammenschließen, stellt hohe Anforderungen an Informationssysteme, wie eine institutionsübergreifende Unterstützung des Versorgungsprozesses und eine starke Integration von Systemen [1].

Diverse Studien [2–5] zur Natur und Vermeidbarkeit von Fehlern in der Medizin belegen den Stellenwert von Gesundheitsinformationssystemen zur Unterstützung von Ärzten und zur Verbesserung der Versorgungsqualität. Fehler treten häufig auf, sind oft aber vermeidbar. Die Ursachen sind nicht in den Fehlern selbst, sondern bereits in den zugrunde liegenden Systemen und Strukturen zu suchen. Einen hohen Anteil besitzen Unterlassungsfehler, als Hauptursachen

konnten die hohe Komplexität in der Medizin in Verbindung mit mangelnder Informationsverfügbarkeit identifiziert werden [6]. Vor allem bei einem Wechsel der behandelnden Institution treten starke Defizite im Informationsfluss auf, wodurch in verteilten Behandlungsprozessen ein hohes Gefahrenpotential liegt. Besonders im Versorgungsnetz ist daher ein starkes Potential für Verbesserungen durch Informationssysteme gegeben. [7]

Die aktuelle IT-Infrastruktur besteht allerdings aus einer Vielzahl unabhängiger und oft inkompatibler Systeme, die die Diskontinuität unterstützen [1]. Ziel des DFG-geförderten Projekts „Grundlagen prozessorientierter Informationssysteme im Krankenhaus und im Versorgungsnetz“ ist es daher, eine geeignete Architektur für ganzheitliche Informationssysteme zur prozessorientierten Unterstützung des Versorgungsnetzes zu erstellen. Im Folgenden wird die im Rahmen dieses Projekts durchgeführte Analyse der Anforderungen an ein Informationssystem von der Seite der Anwender beschrieben, von der auf die technischen und funktionalen Anforderungen an eine solche Architektur abstrahiert wird. Als zentrale Problemstellung tritt die Integration inkompatibler Systeme in den Vordergrund, die verschiedene Formen von Lösungsansätzen erfordert.

## 1.2 Kommunikationsstandards und verwandte Projekte

Die Datenkommunikation in der Medizin erfolgt heute nach verschiedenen nachrichtenbasierten Standards. Im klinischen Bereich ist HL7 („Health Level 7“) weit verbreitet. Dazu kommen DICOM („Digital Imaging and Communications in Medicine“) und EDIFACT („Electronic Data Interchange For Administration, Commerce and Transport“). Arztpraxen verwenden jedoch andere Standards wie die xDT-Familie, z. B. BDT („Befunddatenträger“) oder LDT („Labordatenträger“). Die unterschiedlichen Standards erschweren die Kommunikation zwischen Kliniken und Praxen deutlich.

Die Menge der maßgeblichen Standards unterliegt einer laufenden Veränderung. Viele Formate werden derzeit im Hinblick auf XML überarbeitet. Version 3 von HL7 wird ein Referenzmodell (RIM, „Reference Information Model“) umfassen; mit CDA („Clinical Document Architecture“; eine Architektur zur Definition klinischer Dokumente auf verschiedenen Abstraktionsebenen, vgl. Abschn. 4) soll der Übergang von nachrichtenbasierten zu dokumentenbasierten Systemen erfolgen. Das deutsche Projekt SCIPHOX ([www.sciphox.de](http://www.sciphox.de)), das ursprünglich HL7 und xDT verknüpfen sollte, beschäftigt sich mit einer konkreten Ausgestaltung der CDA und hat bereits einige Definitionen für bestimmte Anwendungsfälle erstellt.

Bekanntere deutsche Projekte, die den sicheren Datentransport (weitgehend unabhängig von dessen Inhalt) adressieren, sind VCS und D2D. Darüber hinaus gibt es zahlreiche industrielle Kommunikationsanbieter, die jedoch meist proprietäre Datenformate verwenden. Oft wird bei Vernetzungsprojekten einem schnellen, zielgerichteten Einsatz der Vorzug vor einem ganzheitlichen Vorgehen gegeben.

Vom Ansatz her mit dem hier vorgestellten Projekt verwandt ist das EU-Projekt PICNIC ([picnic.euspirit.org](http://picnic.euspirit.org)), welches ebenfalls versucht, eine flexible

Kommunikationsarchitektur mit austauschbaren Komponenten zu entwerfen und zu implementieren. Dort werden zwei Schichten definiert mit domänenspezifischen und -unabhängigen Komponenten, die auf bestehenden Transportmechanismen beruhen und auf denen Applikationen aufgebaut werden sollen. Allerdings wird nicht diskutiert, welche Anforderungen dem zugrunde liegen, auch die Integrationsproblematik wird nicht näher adressiert.

Verschiedene deutsche Studien bestätigen die Notwendigkeit von Versorgungsnetzen, plädieren aber dabei auf eine gesetzliche Normierung von Standards zur Vereinfachung der Integrationsproblematik [8], wobei die Integration von vorhandenen Systemen oder Standards weitgehend ignoriert wird.

## 2 Anforderungsanalyse

Für die Entwicklung einer IT-Infrastruktur gelten verschiedenste Rahmenbedingungen (technisch, politisch etc.), die teilweise häufigen Änderungen unterworfen sind. Für ein praktikables und akzeptiertes System sind vor allem aber die Anforderungen seitens der Anwender wichtig. Daher wurden in der ersten Projektphase die bestehende Fachliteratur daraufhin untersucht und strukturierte Interviews mit Niedergelassenen und Klinikärzten geführt.

Die Literatur hat sich in den vergangenen Jahren eingehend mit Bedarf und Ansprüchen an Informationssysteme befasst. Als zentrale Problemfelder werden die Integration heterogener Teilsysteme (woraus die Notwendigkeit zu offenen Systemen, aber auch gemeinsamen Ontologien und Standards erwächst), Adaptierbarkeit an unterschiedliche Prozesse und Bedingungen sowie eine Reihe von soziotechnischen Aspekten aufgeführt. Bedarf und Potential werden vor allem im Versorgungsnetz deutlich:

- Entlassbriefe nach einer Klinikbehandlung waren nur in 15% der Nachbehandlungsbesuche [9] und 27% der Fälle [10] verfügbar. Bei Vorliegen wurde ein signifikant niedrigeres Rehospitalisierungsrisiko festgestellt. [9]
- In 26% der Entlassbriefe fehlten wichtige Informationen [11], 36% waren nach [10] nicht korrekt.
- Die Mehrheit der entlassenen Patienten haben noch offene medizinische Probleme. 49% erlebten dabei Kontinuitätsprobleme, die zum Teil zu einem erhöhten Rehospitalisierungsrisiko führten. [12]

Die Ergebnisse der Arztinterviews korrelieren stark mit der Literatur. Als Hauptprobleme werden ein unzureichender Informationsfluss (mangelhafte oder nicht verfügbare Daten, fehlende Vorbereitung, schwierige Kontaktaufnahme) zwischen Klinik und niedergelassenen Ärzten sowie eine schlechte Koordination (vor allem zwischen Institutionen, führt zu langen Wartezeiten und hohen Latenzen) genannt. Darauf aufbauend formulierten die Befragten Ihre Anforderungen an eine IT-Unterstützung. Elementar sind die elektronische Erstellung und Übermittlung von Arztbriefen und Befunden bzw. die Verfügbarkeit von Patientendaten (z. B. elektronische Patientenakte). Weiter hilfreich wären elektronisch unterstützte Auftragserstellung, Terminplanung, Pfade, Disease-Management-Programme und der Zugriff auf Fachwissen.

### 3 Funktionale Aspekte

Aufgrund der hohen Dynamik in der Medizin kann die Liste der Anforderungen nicht vollständig und allgemeingültig sein. Daraus erwachsen aber gerade die im Hinblick auf die Systemarchitektur zentralen Anforderungen *Integrationsfähigkeit*, *Anpassbarkeit* und *Erweiterbarkeit*. Letztere kann über die Kapselung von Funktionalität in Komponenten erreicht werden. Zur Beschränkung des Entwicklungsaufwands sind möglichst generische Komponenten sinnvoll, was eine Schichtenarchitektur aus domänenspezifischen und -unspezifischen Komponenten motiviert.

Zur Identifikation und Dokumentation der Dienste wurden von den Anforderungen ausgehend mit einer Anwendungsfallanalyse typische Kommunikationsmuster identifiziert. Diese enthalten eine strukturierte Beschreibung des Anwendungskontextes ergänzt um ein UML-Aktivitätsdiagramm. Das Abstraktionsniveau ist so gewählt, dass eine hohe Allgemeingültigkeit erreicht wird.

Die Komplexität und Änderungshäufigkeit von Prozessen in Versorgungsnetzen erfordert einfach konfigurierbare Systeme. Zur Spezifikation von Prozessen ist vor allem Wissen der Anwendungsdomäne notwendig, was eine enge Zusammenarbeit zwischen Experten aus Medizin und Informatik in den Entwicklungsprozessen erfordert [13]. Dafür bieten sich graphische Prozessmodelle mit hoher Verständlichkeit und guter Kommunizierbarkeit an. Über ein entsprechendes Modellierungswerkzeug können die eingeführten Kommunikationsmuster an die konkreten Projektbedingungen angepasst und notwendige Informationen ergänzt werden. Das verwendete Modellierungswerkzeug verfolgt dabei einen aspektorientierten Modellierungsansatz, der ursprünglich für Workflow-Managementsysteme konzipiert wurde. Mittels des aspektorientierten Ansatzes [14] können wir flexibel die Datenlogistik in Versorgungsnetzen modellieren und anschließend zur Ausführung in einem XML-Format exportieren und zur Konfiguration unserer Architektur verwenden. Als Basis hierzu dient ein Prozessmetamodell in XML-Schema.

### 4 Integration

Neben den Anwendungsanforderungen muss ein Informationssystem im Versorgungsnetz eine umfassende Integration der vorliegenden heterogenen Systemlandschaft ermöglichen. Dabei sind vor allem Dateninkompatibilitäten auf verschiedenen inhaltlichen Ebenen (vgl. [15]) zu behandeln, die entsprechende Konzepte erfordern.

#### 4.1 Datenintegration

**Syntaktische Kompatibilität** Daten können auf verschiedene Weise repräsentiert werden. Dafür wird eine syntaktische Transformation nötig. Die Verwendung von XML würde hier mit seinen vielseitigen Erweiterungen (wie XMLSchema, XSLT) mächtige Instrumente für Datenmanagement und

-konvertierung zur Verfügung stellen, ohne dass dadurch die Austauschbarkeit oder Plattformunabhängigkeit eingeschränkt würden.

**Ontologische Kompatibilität** Semantische Heterogenität auf Typebene ist bereits in den Anwendungen angelegt, z. B. durch das verwendete Datenbankschema. Eine nachträgliche Anpassung inkompatibler Datenbankschemata ist ein schwieriges Integrationsproblem und oft ohne Eingriffe in die Datenbankschemata nicht lösbar [16, 17]. Diese Inkompatibilität kann nur durch Einigung auf gemeinsame Ontologien beseitigt werden, allerdings existieren bereits viele inkompatible Standards. In Kliniken ist das nachrichtenbasierte HL7, das die Semantik der auszutauschenden Daten definiert, der De-Facto-Standard, wodurch die Integration zum Teil vereinfacht wird. Aber auch hier existieren Inkompatibilitäten durch unvollständige Implementierungen oder unterschiedliche Interpretationen verschiedener Hersteller. Für eine Verknüpfung von HL7 mit dem in Praxen üblichen xDT müssten diese aufeinander abgebildet werden, was wieder dem Mapping verschiedener Datenbankschemata entspricht und wiederum die Problematik unvereinbarer Konzepte mit sich bringt.

**Terminologische Kompatibilität** Während semantische Datenkompatibilität auf Typebene der Kompatibilität von Datenbankschemata entspricht, korrespondiert semantische Kompatibilität auf Instanzebene mit den Datenbankinhalten [18]. Beides tritt auf, die Instanzebene ist jedoch im Grundsatz weniger problematisch. Terminologische Inkompatibilität tritt zur Laufzeit auf, indem Daten unterschiedlicher Semantik in Systeme eingegeben werden. Dies wird typischerweise durch die Verwendung von standardisierten Terminologiekatalogen vermieden, wobei Medical Entity Dictionaries oder Terminologiedienste zur Vermittlung dienen. Die Übersetzung ist eine typische Aufgabe für ETL-Werkzeuge, aber auch Thesauri wie UMLS (Universal Medical Language System) kommen zum Einsatz. Wir planen entsprechende Komponenten auch in unserer Architektur für ein Informationssystem im Versorgungsnetz.

## 4.2 Gestaffelte Integration

Ein Qualitätsmerkmal für ein Informationssystem im Versorgungsnetz ist die Möglichkeit für unterschiedliche Teilnehmer, sich einfach an dieses System anzubinden. Das Vorschreiben einer bestimmten Ontologie wäre daher sehr problematisch. Deshalb werden Kompromisslösungen benötigt, die es erlauben, auch inkompatible Systeme anzubinden. Unser Konzept sieht vor, semantische Kompatibilität auf verschiedenen, differenzierten Abstraktionsstufen zu betrachten. CDA [19] unterstützt dies durch ihre dreistufige, hierarchische Dokumententypspezifikation. Stufe 1 spezifiziert dabei lediglich die Kontextdaten, lässt den Inhalt aber beliebig. Stufe 2 spezifiziert den Dokumenteninhalte auf einem hohen Abstraktionsgrad, indem beispielsweise bestimmte Abschnitte vorgegeben werden. Stufe 3 schließlich spezifiziert detailliert den Inhalt. Damit können CDA-Daten auch bei unbekanntem Ontologien behandelt werden, auch wenn sie dann nur uninterpretiert verarbeitet oder angezeigt werden.

### 4.3 Verknüpfung von Daten

Es bleibt in jedem Fall erforderlich, sich auf eine gewisse Minimalmenge an Kontextdaten zu einigen, um Daten wie klinische Dokumente sicher miteinander verbinden zu können. Dafür sind zentrale Referenzierungsstrukturen wie ein Master Patient Index nötig. Dieser ermöglicht es, die im System enthaltenen Patientendaten zu einer gemeinsamen Patientenakte zusammenzufügen. Die Daten können dabei zentral oder verteilt vorgehalten werden, für den Betrachter stellen sie sich jedoch als (virtuell-)zentrale Patientenakte dar.

## 5 Zusammenfassung

Die zunehmende Veränderung des Gesundheitssystems hin zu Behandlungsprozessen in Versorgungsnetzen erfordert geeignete, prozessunterstützende Informationssysteme, die einen erheblichen Einfluss auf die Prozessqualität nehmen können. Eine innerhalb des hier dargestellten Forschungsprojekts durchgeführte Analyse der Anwendungsanforderungen ergab Erweiterbarkeit, Anpassbarkeit und Integration als zentrale Themen. Hierfür wird eine Komponentenarchitektur mit generischen Diensten entwickelt, die die einfache Änderung und Ergänzung von Komponenten erlauben soll. Für die Datenintegration existieren Lösungen wie Syntaxkonverter und Terminologievermittlungsdienste. Problematisch ist jedoch die semantische Kompatibilität der Typebene, die sich nicht automatisiert handhaben lässt und oft Eingriffe in zugrunde liegende Modelle erfordert. Um eine einfache Fremdsystemintegration zu ermöglichen, müssen auch Systeme unbekannter Ontologien einfach eingebunden werden können. Die CDA geht mit ihrem mehrstufigen Abstraktionskonzept einen Schritt in diese Richtung. In unserem Projekt wird gegenwärtig geprüft, wie sich dies im Rahmen einer komponentenbasierten, erweiterbaren Systemarchitektur einbeziehen lässt, so dass die semantische Kompatibilität schrittweise ausgebaut werden kann. Dabei wird insbesondere auch geprüft, inwiefern verfügbare Gesundheitsplattformen als Basis einer solchen Architektur verwendet werden können. Erste Analysen zeigen, dass viele existierende Plattformen sich nicht oder nur bedingt für den Einsatz im Rahmen einer Schichtenarchitektur eignen, weil sie bereits mit einer Festlegung auf eine spezifische, wenig generische Ontologie verbunden sind.

*Das Forschungsprojekt „Grundlagen prozessorientierter Informationssysteme im Krankenhaus und im Versorgungsnetz“ wird unterstützt von der Deutschen Forschungsgemeinschaft (DFG).*

## Literatur

1. Kuhn, K.A., Giuse, D.A.: From hospital information systems to health information systems. Problems, challenges, perspectives. *Methods Inf Med* **40** (2001) 275–287

2. Brennan, T.A., Leape, L.L., Laird, N.M., Hebert, L., Localio, A.R., Lawthers, A.G., Newhouse, J.P., Weiler, P.C., Hiatt, H.H.: Incidence of adverse events and negligence in hospitalized patients. Results of the Harvard Medical Practice Study I. *N Engl J Med* **324** (1991) 370–376
3. Leape, L.L., Brennan, T.A., Laird, N., Lawthers, A.G., Localio, A.R., Barnes, B.A., Hebert, L., Newhouse, J.P., Weiler, P.C., Hiatt, H.: The nature of adverse events in hospitalized patients. Results of the Harvard Medical Practice Study II. *N Engl J Med* **324** (1991) 377–384
4. Bhasale, A.: The wrong diagnosis: identifying causes of potentially adverse events in general practice using incident monitoring. *Fam Pract* **15** (1998) 308–318
5. Wilson, R.M., Harrison, B.T., Gibberd, R.W., Hamilton, J.D.: An analysis of the causes of adverse events from the quality in australian health care study. *Med J Aust* **170** (1999) 411–415
6. Leape, L.L.: A systems analysis approach to medical error. *J Eval Clin Pract* **3** (1997) 213–222
7. Bates, D.W., O’Neil, A.C., Boyle, D., Teich, J., Chertow, G.M., Komaroff, A.L., Brennan, T.A.: Potential identifiability and preventability of adverse events using information systems. *J Am Med Inform Assoc* **1** (1994) 404–411
8. BITKOM / VDAP / VHitG / ZVEI: Einführung einer Telematik-Architektur im deutschen Gesundheitswesen – Expertise. [http://www.ztg-nrw.de/down/262/telematik\\_expertise.pdf](http://www.ztg-nrw.de/down/262/telematik_expertise.pdf) (2003)
9. van Walraven, C., Seth, R., Laupacis, A.: Dissemination of discharge summaries. Not reaching follow-up physicians. *Can Fam Physician* **48** (2002) 737–742
10. Wilson, S., Ruscoe, W., Chapman, M., Miller, R.: General practitioner-hospital communications: A review of discharge summaries. *J Qual Clin Pract* **21** (2001) 104–108
11. van Walraven, C., Weinberg, A.L.: Quality assessment of a discharge summary system. *CMAJ*. **152** (1995) 1437–1442
12. Moore, C., Wisnivesky, J., Williams, S., McGinn, T.: Medical errors related to discontinuity of care from an inpatient to an outpatient setting. *J Gen Intern Med* **18** (2003) 646–651
13. Lenz, R., Kuhn, K.A.: Towards a continuous evolution and adaptation of information systems in healthcare. *Int J Med Inf* (2003) In Press.
14. Jablonski, S., Bußler, C.: Workflow management – modeling concepts, architecture and implementation. International Thomson Computer Press, London (1996)
15. Lenz, R., Kuhn, K.A.: Intranet meets hospital information systems: the solution to the integration problem? *Methods Inf Med* **40** (2001) 99–105
16. Colomb, R.M.: Impact of semantic heterogeneity on federating databases. *The Computer Journal* **40** (1997) 235–244
17. Elmagarmid, A., Rusinkiewicz, M., Sheth, A., eds.: Management of Heterogeneous and Autonomous Database Systems. Morgan Kaufmann Publishers, San Francisco, California (1999)
18. Musen, M.A.: Domain ontologies in software engineering: use of Protege with the EON architecture. *Methods Inf Med* **37** (1998) 540–550
19. Dolin, R.H., Alschuler, L., Beebe, C., Biron, P.V., Boyer, S.L., Essin, D., Kimber, E., Lincoln, T., Mattison, J.E.: The HL7 Clinical Document Architecture. *J Am Med Inform Assoc* **8** (2001) 552–569



# Integration in der medizinischen Anwendung: Prozessbasierte Datenlogistik<sup>1</sup>

Sascha Müller, Rainer Lay, Christian Meiler, Stefan Jablonski

Lehrstuhl für Informatik 6 (Datenbanksysteme)  
Friedrich-Alexander-Universität Erlangen-Nürnberg  
Martensstr. 3, D-91058 Erlangen  
{sascha.mueller, rainer.lay, christian.meiler, stefan.jablonski}@informatik.uni-erlangen.de

**Zusammenfassung.** Die möglichst optimale Integration klinischer Anwendungssysteme ist vital für den modernen Klinikbetrieb. Wir beschreiben einen Ansatz zur prozessorientierten Datenlogistik, der es erlaubt, das medizinische Personal entlang dem Behandlungsprozess mit notwendigen Daten zu versorgen. Durch Transformation graphischer Prozessmodelle in eine datenzentrierte Darstellung kann eine automatisierte Konfiguration von Kommunikationsservern erreicht werden. Im Vordergrund steht dabei die Datenversorgung der am Behandlungsprozess beteiligten Personen und nicht die direkte Unterstützung des Prozesses, wie sie durch Workflow-Management-Systeme erreicht wird. Der Ansatz wird in einer Fallstudie vorgestellt.

## 1 Einleitung

In der klinischen Datenverarbeitung besteht die Notwendigkeit zur Integration verschiedenster Anwendungssysteme mit dem Ziel existierende und zukünftige Behandlungsprozesse optimal zu unterstützen. Zwei grundlegend verschiedene Ansätze dominieren den Markt: Vollständig integrierte Krankenhausinformationssysteme (KIS), die mit einem holistischen Ansatz die Schnittstellenproblematik weitgehend zu vermeiden versuchen. Als Alternative existieren vor allem Kommunikationsserver-basierte Lösungen, die mit standardisierten Schnittstellen wie HL7 („Health Level 7“) [1] den Datenaustausch zwischen Einzelsystemen ermöglichen.

Im Fall der Kommunikationsserver stellt sich die Frage, wie die Datenflüsse optimal gestaltet werden können, um jederzeit hohe Datenverfügbarkeit zu gewährleisten und gleichzeitig die Prozesskosten zu minimieren. Eine Lösung für diese Problematik könnte eine prozessorientierte Vorgehensweise sein, welche beispielsweise die Datenverteilung entlang den Behandlungsstationen eines Patienten organisiert. Die Vorteile einer derartigen prozessbasierten Datenlogistik sind offensichtlich:

---

<sup>1</sup> Der Sonderforschungsbereich 539: "Glaukome einschließlich Pseudoexfoliations-Syndrom (PEX)" wird unterstützt von der Deutschen Forschungsgemeinschaft (DFG).

- Konfigurationen für Kommunikationsserver können automatisch aus umfassenden Prozessmodellen generiert werden.
- Klinische Pfade können abgebildet und fallbasiert unterstützt werden.
- Vorhandene Infrastruktur kann effizienter genutzt werden, da sich die Lücke zwischen Anwendungswissen und der technischen Umsetzung, beispielsweise der Konfiguration eines Kommunikationsservers, schließt.

Wir beschreiben im Folgenden einen Ansatz, der die genannten Vorteile der prozessbasierten Datenlogistik im Krankenhausbereich umsetzt. Dazu betrachten und bewerten wir im nächsten Abschnitt Basistechnologien wie Kommunikationsserver und Workflow-Management. Abschnitt 3 beschreibt unseren prozessbasierten Lösungsansatz und Abschnitt 4 illustriert schließlich mit einer Fallstudie unsere Methodik.

## 2 Basistechnologien

Die Basistechnologien Kommunikationsserver und Workflow-Management-Systeme (WfMS) werden im Folgenden kurz eingeführt und im Anwendungskontext des hier vorgestellten Ansatzes bewertet.

### 2.1 Kommunikationsserver

Kommunikationsserver sind eine in Krankenhäusern weit verbreitete Middleware (MOM, Message-Oriented-Middleware) und ermöglichen den Datenaustausch zwischen den verschiedenen medizinischen Systemen. Durch die Definition von Schnittstellen auf Basis von HL7 bilden Kommunikationsserver eine zentrale Plattform zum asynchronen und quitierten Datenaustausch. Vorteile sind unter anderem die deutlich reduzierte Schnittstellenanzahl zwischen den einzelnen Systemen und das standardisierte Nachrichtenformat HL7. Ein Kommunikationsserver erfüllt dabei im Wesentlichen die drei folgenden Hauptaufgaben [2]: Empfang von Datenpaketen, semantische und syntaktische Transformation vom Quell- in das Zielformat und Auslieferung des Datenpakets.

Die Konfiguration erfolgt dabei regelbasiert und oft auf niedrigem Abstraktionsniveau. Für umfangreiche heterogene Netzwerke ergibt sich so schnell eine unübersichtliche Konfiguration, die schwer zu überblicken und verifizieren ist. Entsprechend aufwändig und fehleranfällig gestaltet sich die Wartung. Wir bezeichnen dieses Vorgehen im Folgenden als Konfiguration auf Instanzebene. Diese bringt eine Reihe von Problemen mit sich: Es existiert eine semantische Lücke zwischen Anwendungsdomäne und der tatsächlichen Implementierung. Ohne die Notwendigkeit derartiger Konfigurationen in Frage zu stellen, sehen wir große Vorteile durch Modellierungskonzepte auf höherem Abstraktionsniveau.

## 2.2 Workflow Management

Workflow-Management-Systeme werden in verschiedensten Anwendungsbereichen eingesetzt und bieten eine umfassende Unterstützung von Anwendungsprozessen. Die Implementierung einer Workflow-Management-Anwendung erfolgt in zwei Schritten: In einem ersten Schritt muss ein Workflow-Modell erstellt und validiert werden. In einem zweiten Schritt wird ein solches Workflow-Modell zur Ausführungszeit instanziiert. Die Anwender interagieren mit dem System über eine Arbeitsliste, welche den am Prozess beteiligten Personen eine Liste der zur Ausführung bereitstehenden Aufgaben präsentiert [3].

Der Einsatz von WfMSen in Krankenhäusern hat eine Reihe von Vorteilen: WfMSe garantieren die koordinierte Ausführung von Prozessen mit Beteiligung von Anwendern und Systemen. Insbesondere der Datenaustausch zwischen den beteiligten Systemen wird organisiert, was WfMSe für den Einsatz im klinischen Umfeld prädestiniert. Ein oft genannter Vorteil von WfM ist die Eigenschaft, alle notwendigen Daten bei der Ausführung eines Prozessschrittes zur Verfügung zu stellen. Gerade in Kliniken kann dieses Merkmal aber auch sehr schnell ins Gegenteil umschlagen, wenn beispielsweise ein medizinischer Notfall die Abweichung vom vorgegebenen Workflow erzwingt. In diesem Fall muss das WfMS umgangen werden und es zeigt sich, dass sehr genau überlegt werden muss, welche Prozesse überhaupt mit Hilfe eines WfMS verwaltet werden können.

Durch die strenge und unflexible Ausführung von Prozessen ergibt sich eine Reihe von Nachteilen. Was in Anwendungsgebieten mit einer überschaubaren Anzahl weitgehend statischer Abläufe, wie beispielsweise dem Banken- oder Versicherungswesen ein gewichtiger Vorteil ist, wird im medizinischen Anwendungsbereich zur Achillesferse, da hier mit zahlreichen hochgradig variierenden Prozessen zu rechnen ist. Die resultierende Variantenvielfalt ist mit WfMS in der Praxis kaum zu erreichen, da diese auf eine strikte Ausführung der vorgegebenen Prozesse ausgelegt sind und die Mächtigkeit der Modellierungssprache im Allgemeinen zu gering ist. Ansätze zur Flexibilisierung der Workflow-Ausführung (z.B. [4]) sind zwar vorhanden, erhöhen aber auch die Komplexität des resultierenden Workflow-Modells.

## 2.3 Bewertung

Sowohl Kommunikationsserver als auch WfMSe bieten Vor- und Nachteile beim Einsatz in Kliniken. WfMSe bieten die notwendigen Modellierungsmittel, um klinische Prozesse auf einem sinnvollen Abstraktionslevel umfassend zu beschreiben. Es zeigte sich aber auch, dass WfMSe oftmals auf einem zu restriktiven Ausführungsmodell basieren und die notwendige Flexibilität nicht liefern können.

Kommunikationsserver bieten aufgrund der verwendeten Konfiguration auf Instanzebene keine problemspezifische Modellsicht, woraus schlechte Wartbarkeit und Lesbarkeit resultiert. Ihre Stärken liegen in dem einfachen und standardisierten Aufbau von Kommunikationsbeziehungen zwischen klinischen Systemen. Ausgehend von diesen Beobachtungen ist es wünschenswert, ein Konzept zu entwickeln, welches die Vorteile von beiden Ansätzen vereint und die Nachteile nicht übernimmt. Unser Konzept der prozessorientierten Datenlogistik folgt dieser Forderung.

### 3 Unser Lösungsvorschlag

Das vorliegende Konzept vereint die Vorteile von WfM und Kommunikationsservern und bietet eine Lösung für den Bereich der klinischen Kommunikation. Folgende Kernforderungen werden dabei erfüllt: Datenverfügbarkeit bei Ausführung eines Prozessschritts, abstrakte Beschreibung der klinischen Prozesse durch Prozessmodelle und Verwendung der Datenaustauschmechanismen von Kommunikationsservern (oder ähnlichen Ansätzen).

Die Ergebnisse aus Abschnitt 2 lassen uns zwei Arten der Unterstützung von klinischen Prozessen erkennen: Modellierungs- und Kommunikationsunterstützung. Dies veranlasst uns zur Definition von zwei Ebenen: Der *Modellierungsebene*, welche eine abstrakte aber umfassende Definition der Kommunikationsbeziehungen enthält und der *Kommunikationsebene*, welche diese Spezifikationen umsetzt. WfMS haben konzeptionelle Stärken auf der Modellierungsebene und können dort verwendet werden. Zusätzlich kommen Teilkomponenten wie die Workflow-Ausführung auch in der Kommunikationsebene zur Geltung. Kommunikationsserver sind ein typischer Vertreter der Kommunikationsebene, aber auch andere Implementierungen können hier angesiedelt werden, wie beispielsweise der adaptive Replikationsmanager [5].

Wir bezeichnen unser Verfahren um klinische Prozesse zu modellieren und auszuführen als *Prozessbasierte Datenlogistik (PDL)*. Es fußt auf den Konzepten des WfM, verzichtet jedoch auf die Ausführung, indem es sich auf die Ableitung von Kommunikationsregeln aus einem Prozessmodell beschränkt. PDL unterscheidet sich weiterhin vom strikten Modell des Workflow Management, indem es unauffällig im Hintergrund dem Anwender die für seine nächsten Schritte benötigten Daten bereitstellt; eine Interaktion mit dem Anwender ist nicht notwendig. Der Anwender behält so volle Handlungsfreiheit und profitiert zugleich von hoher Datenverfügbarkeit. Folgende Schritte charakterisieren PDL:

1. *Die klinischen Prozesse müssen aufgenommen und in einem Workflow-Modell dokumentiert werden.*

Analog zum Workflow-Management steht am Anfang eine umfassende Prozessaufnahme, die auf verfügbaren Informationen basiert, wie z.B. medizinische Leit- und Richtlinien, medizinisches Personal, EDV-Systemlandschaft, etc.

2. *Das Workflow-Modell muss in ein datenzentriertes Modell umgewandelt werden.*

Mit dieser Transformation werden die beteiligten Systeme und die zwischen ihnen ausgetauschten Daten in den Mittelpunkt der Betrachtung gestellt. Man kann daraus sehr leicht die Regeln zur Beschreibung der Datenlogistik ableiten. Ein Workflow-Modell wird in ein datenzentriertes Modell gewandelt (siehe Abb. 1), indem aus jedem Datenfluss zwischen zwei Prozessschritten ein Datenlogistikschritt erzeugt wird. Dieser Datenlogistikschritt benötigt die ausgehenden Datencontainer des vorhergehenden Workflow-Schritts und die eingehenden Datencontainer des nachfolgenden Workflow-Schritts. Zusätzlich werden noch die Systeme der beteiligten Workflow-Schritte benötigt, da diese den Speicherort der zu transportierenden Daten spezifizieren. In unserem Beispiel (vgl. Abb. 1) wird aus den Schritten „record glaucoma data“ und „store diagnostic data“ ein Datenlogistikschritt, der die Daten „IOP“ und „Blood Pressure“ von „Telephone

System“ zu „Glaucoma Register“ transportiert. Die Problematik der Datentransformation bedingt durch unterschiedliche Formate, Terminologien und Ontologien in Quell- und Zielsystem ist hier nicht näher betrachtet. Eine Beschreibung der Datencontainer in dieser Hinsicht ist jedoch vorgesehen.

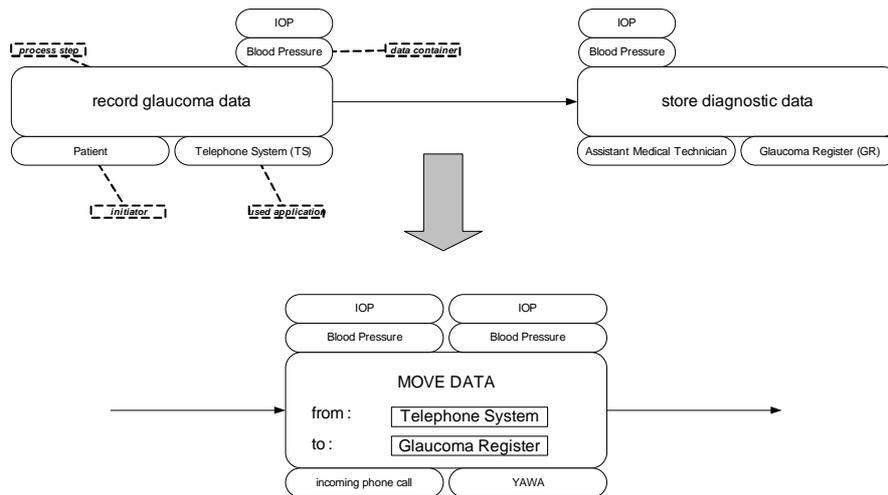


Abb. 1. Die Abbildung eines Workflow-Schrittes in einen Datenlogistikschritt

### 3. Die Datenlogistikprozesse müssen ausgeführt werden.

Für diese Aufgabe können verschiedene Techniken zum Einsatz kommen. Denkbar wären auch WfMSe mit leichten Modifikationen. Es bieten sich im klinischen Umfeld besonders Kommunikationsserver an, für die automatisch eine Konfiguration aus dem datenzentrierten Modell erzeugt werden kann.

Die Hauptvorteile der PDL liegen in der Wiederverwendung bekannter und bewährter Techniken und in der erheblich vereinfachten und umfassenderen automatischen Konfiguration von Systemen der Kommunikationsschicht, wie beispielsweise Kommunikationsserver.

## 4 Fallstudie: Vom Workflow-Modell zur prozessorientierten Datenlogistik

In diesem Abschnitt beschreiben wir, wie man ein Workflow-Modell in eine gültige Konfiguration für ein System der Kommunikationsebene, in unserem Beispiel ein Wrapper-Framework, transformiert. Der verwendete, vereinfacht dargestellte Prozess (Abb. 2) entspringt einem Projekt aus dem Sonderforschungsbereich 539 der Universitätsaugenklinik Erlangen-Nürnberg.

## Prozess: Self-Tonometry

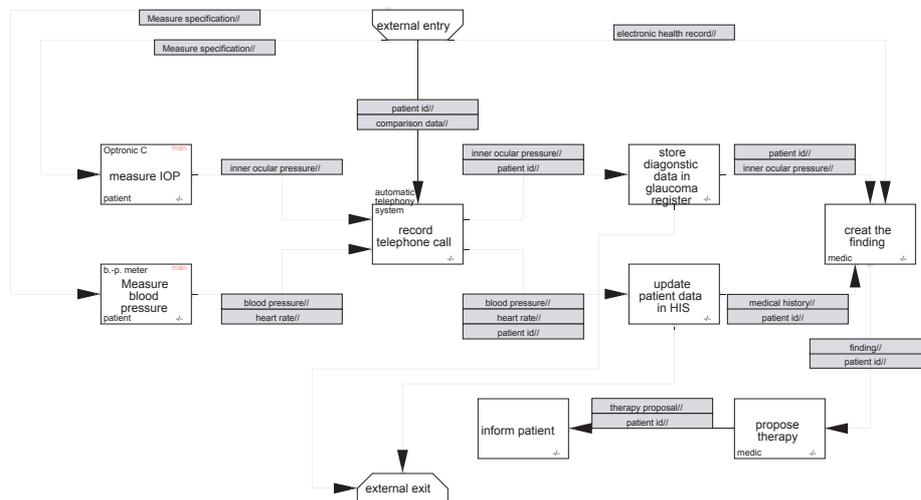


Abb. 2. Der Selbsttonometrie Prozess

Kurz zum medizinischen Hintergrund: Einige Glaukomatpatienten (Grüner Star) müssen den Augeninnendruck (IOP, inner ocular pressure) über einen längeren Zeitraum selbstständig messen. Um den Ablauf zu optimieren, können die Patienten ihre Daten mit Hilfe eines automatischen Telefonesystems an die Klinik übermitteln. Dabei muss der Patient sich authentifizieren und dann seine Daten eingeben. Vom automatischen Telefonesystem werden die Daten an das KIS (Abrechnung) und an die zentrale Glaukomdatenbank (Forschung) weitergeleitet. Ein Augenarzt überprüft regelmäßig die Daten und verordnet eine Therapie oder eine weitere Untersuchung.

In diesem vereinfachten Beispiel soll die Datenübertragung des automatischen Telefonesystems in das KIS und das zentrale Glaukomregister unter Verwendung der PDL Methodologie konfiguriert werden.

Das abgebildete Prozessmodell ist nicht nur eine graphische Repräsentation des Prozesses. Das verwendete Modellierungswerkzeug (I>PM [6]) erlaubt eine umfassende, aspektorientierte Beschreibung mit allen für die PDL notwendigen Daten.

Um die Beschreibung für die Konfiguration verwenden zu können, wird ein XML Export des Modells erzeugt. Dieser wird mit Hilfe eines XSLT Skripts in ein passendes datenzentriertes XML Modell gewandelt. Im folgenden Ausschnitt ist ein entsprechend transformierter Prozessschritt zu sehen.

```
<PEK xsi:type="process" Name="store diagnostic data in glaucoma register"
ID="REP_ORTR(12675)" >
  <Attribut Name="ParentID" Wert="REP_ORTP(8348)" />
  <data_container_out Name="inner_ocular_pressure" ID="REP_ORTP(8350)" />
  <data_container_out Name="blood_pressure" ID="REP_ORTP(8318)" />
</PEK>
```

Um von dieser Beschreibung zu einer gültigen Konfiguration für das verwendete Wrapper-Framework YAWA („Yet Another Wrapper Framework“) [7] zu kommen, sind folgende Schritte notwendig: 1. Lokalisierung, d.h. die Auswahl der verwendeten

Systeme, 2. Erzeugen der Konfiguration für das Wrapper Framework (eine Teilkonfiguration ist im folgenden Ausschnitt zu sehen), 3. Erzeugung von ECA Regeln.

```

<EXTRACTOR><VIEWS>
  <pressure>SELECT * FROM All_Pressure_Data</pressure>
</VIEWS></EXTRACTOR>
<RESULTPROCESSOR>
  <TARGETS><IOP> <TARGET_RELATION>IOP</TARGET_RELATION>
  <ATTRIBUTES>
    <ID TYPE="Integer" />
    <Exam_Date TYPE="Date" FORMAT="yyyy-MM-dd" />
    <Patient_ID TYPE="String" />
    ..
  </ATTRIBUTES>
</IOP></TARGETS>
</RESULTPROCESSOR>

```

Anhand dieser Fallstudie konnte exemplarisch gezeigt werden, dass durch eine Änderung des Workflow-Modells automatisiert eine Anpassung der Konfiguration des Kommunikationssystems erfolgen kann.

## 5 Fazit

Dieses Papier präsentiert einen Ansatz zur prozessbasierten Konfiguration des Datenaustausches zwischen klinischen Systemen. Ziel ist dabei eine Versorgung des medizinischen Personals mit relevanter Information entlang des Behandlungsprozesses. Aus graphisch modellierten Prozessmodellen werden hierzu Datenlogistikprozesse abgeleitet. Eine Fallstudie verdeutlicht den Ansatz.

## Literatur

- 1 Hammond, W. E., De Moore, G. J. E. (Hrsg.) et al.: "HL7: A Protocol for the Interchange of Healthcare Data.". Progress in Standardization in Health Care Informatics. IOS Press, 1993.
- 2 Lange M., Prokosch H., Hasselbring W., 1999, Eine Taxonomie für Kommunikationsserver im Krankenhaus. Informatik, Biometrie und Epidemiologie in Medizin und Biologie 30 (1), 21-34, 1/1999.
- 3 Jablonski, S.; Bußler, C.: Workflow management - modeling concepts, architecture and implementation. London. International Thomson Computer Press, 1996.
- 4 Reichert M.; Dadam P: A Framework for Dynamic Changes in Workflow Management Systems. DEXA Workshop 1997.
- 5 Niemann H., Hasselbring W.: Adaptive Replikationsstrategie für heterogene, autonome Informationssysteme. In Tagungsband zum 14. GI-Workshop Grundlagen von Datenbanken, 2002.
- 6 ProDatO GmbH, i>ProcessManager, <http://www.prodato.de/software/processmanager>, Retrieved 2003-12-03.
- 7 Lang, M.; Lay, R.: Yawa - Yet Another Wrapping Architecture. BTW 2003.



# EAI und ITIL im Krankenhaus – was kosten die IT-Dienstleistungen?

Andreas Zimolong<sup>1</sup>, Gregor Nöthen<sup>1</sup>, Stefan Fritz<sup>2</sup>, Sylvia Thun<sup>1</sup>, Kurt Becker<sup>1</sup>

<sup>1</sup> synaix Beratung für das Gesundheitswesen, Im Süsterfeld 1, 52072 Aachen, Germany  
{azimolong, gnoethen, sthun, kbecker}@synaix-healthcare.de

<sup>2</sup> eliqos Business Consulting & Management GmbH, Im Süsterfeld 1 52072 Aachen, Germany  
sfritz@eliqos.de

**Zusammenfassung.** Neben der Homogenisierung der IT-Systemlandschaft durch ein integrierendes Krankenhausinformationssystem ist es auch notwendig, eine entsprechende Organisationsstruktur für den Betrieb der EDV des Krankenhauses zu erreichen. Dieses hat zum einen das Ziel, verfügbare Ressourcen effizient einzusetzen. Zum anderen müssen kostentransparente Schnittstellen zu externen Anbietern spezifischer IT-Dienstleistungen geschaffen werden, da hochspezialisiertes Wissen von der EDV-Abteilung eines Krankenhauses in der Regel nicht kostengünstig vorgehalten werden kann. Hierfür bietet sich eine Organisation entsprechend dem ITIL-Modell an, welche EDV auf die Erbringung von IT-Dienstleistungen ausrichtet. Während die Organisation der EDV-Abteilung nach ITIL weniger Schwierigkeiten bereitet, zeigt es sich, dass vielfach IT-Dienstleister nicht oder nur unzureichend auf die Bereitstellungen von IT-Dienstleistungen auf der Basis von vereinbarten Service-Levels vorbereitet sind. Eine Aufteilung von Verantwortlichkeiten auf der Grundlage von Leistungsscheinen erfordert noch Klärungsbedarf. Die Anbieter sind an der Übernahme von IT-Dienstleistungen interessiert, so dass sich diese Form der Organisation auch in kleineren EDV-Abteilungen vermehrt durchsetzen wird.

## 1. Einführung

Die Krankenhäuser in Deutschland stehen vor der Einführung eines vollständigen Fallpauschalensystems für die Abrechnung der Patientenbehandlung. Dies erfordert sehr kurzfristig die Schaffung effizienter Dokumentations- und Kommunikationsstrukturen in und zwischen den verschiedenen Arbeitsbereichen des Krankenhauses. Um zukünftig wettbewerbsfähig medizinische Leistungen zu erbringen und dafür auch leistungsgerechte Erlöse zu erzielen, ist es von besonderer Wichtigkeit, die abrechnungsrelevanten Behandlungsdaten direkt in den Behandlungsprozessen zu erfassen. Mittelfristig wird die EDV auch steuernde („anleitende“) Aufgaben in den Behandlungsprozessen übernehmen und damit zu mehr Effizienz und Wirtschaftlichkeit in den Behandlungsprozessen führen. Zentrale Bedeutung kommt dabei einem integrierenden Krankenhausinformationssystem (KIS) zu, welches allen relevanten Bereichen des Krankenhauses Zugang zu den für die Aufgabenerfüllung notwendigen Informationen bietet. Dabei übernimmt das KIS die Funktion einer vereinheitlichenden Plattform, welche die heterogene IT-Systemlandschaft des

Krankenhaus zu einem umspannenden Informations- und Kommunikationsmedium verbindet. Insbesondere die Schnittstelle zwischen dem administrativen und dem medizinisch/pflegerischen System erlangt dabei besondere Bedeutung für die Sicherung des wirtschaftlichen Überlebens des Krankenhauses [1]. Dabei ist es auch von großer Bedeutung, dass diese und andere Schnittstellen vereinheitlicht und auf der Basis gängiger Standards (HCM, BAPI, HL/7) realisiert werden [2].

## 2. Methoden

Neben der reinen technischen Installation eines Krankenhausinformationssystems ist die organisatorische Integration der Arbeitsabläufe notwendig, um den erwarteten Nutzen der Investition zu realisieren. „EAI ohne Geschäftsprozessintegration ist eine Sackgasse“ [3]. Falls dies bei der Umstrukturierung der EDV nicht berücksichtigt wird, kann die EDV – Einführung zu einer für alle Beteiligten bedrohlichen Investitionsruine werden. Daher empfiehlt sich für die Planung der EDV eine strukturierte Vorgehensweise mit der frühzeitigen Klärung der einzelnen Rahmenbedingungen.



Abbildung 1: Rahmenkonzept für EDV – Organisationsprojekte im Krankenhaus

Bei synaix Beratung für das Gesundheitswesen wurde ein Modell für EDV-Einführungen in Krankenhäusern entwickelt, welches sich an üblichen Informatikstandards orientiert [4]. Mit diesem Modell lassen sich komplexe EDV-Projekte strukturiert und übersichtlich planen und durchführen (vgl. Abbildung 1).

### 2.1. Ressourcenstrategie

Grundlegende Bedeutung kommt dabei der Ressourcenstrategie zu. So erfordert die EDV im Krankenhaus von den Beteiligten eine Vielzahl qualifizierter Tätigkeiten, angefangen bei der Netzwerktechnik, über die Datenbankadministration bis zur Betreuung der verschiedenen Anwendungssysteme. Insbesondere kleinere bis mittlere Krankenhäuser können es sich nicht leisten, Mitarbeiter mit allen erforderlichen

Qualifikationen einzustellen und entsprechend aus- und weiterzubilden. Das Verhältnis von betreuten Arbeitsplätzen und EDV-Mitarbeitern lässt sich verbessern, wenn sich Krankenhäuser zusammenschließen und ihre EDV – Aktivitäten bündeln. Ist eine Zusammenarbeit nicht möglich oder ergibt sich auch durch den Zusammenschluss nicht die erforderliche kritische Masse für die Bildung einer EDV-Abteilung, so müssen spezifische Dienstleistungen von extern eingekauft werden. Dies gilt insbesondere für Themengebiete, welche ein sehr spezialisiertes Wissen erfordern und im täglichen EDV-Betrieb des Krankenhauses selten zum Einsatz kommen. Als Beispiel sei hier das Einrichten eines Kommunikationsservers genannt oder der sichere Betrieb von Terminalservern und Firewalls. Dabei ist zu beachten, dass die Aufgabenteilung zwischen einem externen Anbieter und der EDV-Abteilung des Krankenhauses sehr unterschiedliche Formen annehmen kann. So ist es in dem o.g. Beispiel sehr wohl denkbar, dass Teile des Betriebs der Firewall, wie z.B. das Überprüfen der Log-files und das Ändern von Filterparametern, durch die EDV des Krankenhauses erfüllt wird, während komplexere Aufgaben vom externen Anbieter wahrgenommen werden. Des Weiteren können auch urlaubs- und krankheitsbedingte Ausfälle vom externen Dienstleister erfüllt werden, so dass für das Krankenhaus der Zwang zur Beschäftigung zueinander redundanter Mitarbeiter entfällt.

## **2.2. Organisationsstrategie**

Um eine reibungslose Zusammenarbeit zwischen der EDV des Krankenhauses und einem externen Dienstleister zu gewährleisten, ist ein gemeinsames Organisationskonzept nach einheitlichen Standards notwendig. Als gemeinsame Grundlage können Best Practice-Erkenntnisse aus dem ITSM (IT Service Management) nach ITIL (IT-Infrastruktur Library [5]) und der Layer-Struktur komplexer IT-Dienste herangezogen werden. Dadurch erfolgt die Schaffung definierter Schnittstellen für die Auslagerung einzelner Serviceleistungen an Dritte, sowie eine Kostentransparenz für die vorgehaltenen und erbrachten IT-Dienstleistungen. Letzteres ist insbesondere vor dem Hintergrund stetig wachsender Budgetanforderungen seitens der Krankenhaus-EDV notwendig.

Die Grundidee der Organisationsstruktur einer derart aufgebauten EDV ist dabei die konsequente Ausrichtung auf die Erbringung von IT-Dienstleistungen. So wird jede Applikation, welche den Anwendern im Krankenhaus zur Verfügung gestellt wird, als eine von der EDV zu erbringende und vom Krankenhaus zu bezahlende Dienstleistung verstanden. Neben diesen Anwender-Dienstleistungen gibt es auch Infrastruktur-Dienstleistungen, welche die Nutzung der Anwendungen erst möglich machen. So muss beispielsweise ein Netzwerk betrieben und gegen unberechtigte Zugriffe von Außen und Innen geschützt werden.

Für jede Dienstleistung werden Service-Level vereinbart, welche insbesondere die Verfügbarkeit der Dienstleistung sowie die Entstörung quantifizieren. Auf der einen Seite gibt dies dem Krankenhaus die Möglichkeit, die mit dem Einsatz von Applikationen verbundenen Ausfallrisiken zu quantifizieren, zum anderen hilft dies der EDV-Abteilung bei der Ressourcen- und Organisationsplanung.

Prominentestes Merkmal einer Serviceorganisation nach ITIL wird für die Anwender die Einrichtung eines zentralen Service-Desks sein (vgl. Abbildung 2). Hier werden zentralisiert die Anfragen der Anwender entgegengenommen zunächst im

Rahmen des Vorfallmanagements bearbeitet [5]. Im Vergleich zu einer „normalen“ Rufbereitschaft der EDV muss die Behebung des Vorfalls innerhalb einer definierten Zeitspanne (beispielsweise 4 Minuten) abgeschlossen sein, das heißt den Anwendern wird innerhalb dieser Zeitspanne eine Lösung ihres Problems oder eine Behelfsstrategie angeboten. Können Vorfälle nicht innerhalb der vorgegebenen Zeitspanne geklärt werden, so erfolgt eine Eskalation zum Problemmanagement in Form eines Trouble-Tickets. Hier wird das Problem von einem in diesem Bereich spezialisierten Mitarbeiter bearbeitet. Dem Trouble-Ticket System steht dabei die wichtige Aufgabe der Steuerung der Arbeitsabläufe zu (vgl. Abbildung 2).

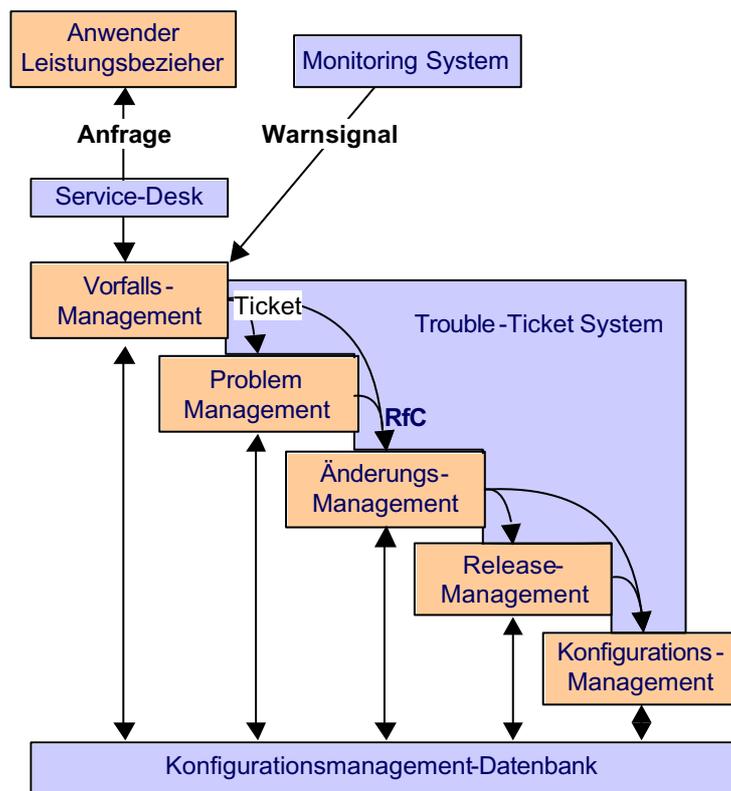


Abbildung 2: Organisation Service-Desk und Einordnung ITSM-Werkzeuge [1]

Diese Unterscheidung in unterschiedliche Stationen der Problembearbeitung und -Lösung ist insbesondere für die Mitarbeiter kleinerer EDV-Abteilungen, wie sie in der Regel in den Krankenhäusern vorherrschen, wichtig: Während sie als Mitarbeiter im Vorfallsmanagement das Ziel haben, dem betroffenen Anwender möglichst schnell eine Fortführung seiner Tätigkeiten zu ermöglichen, müssen sie als Mitarbeiter des Problemmanagements die Ursachen des Fehlers ergründen. Auf diese Weise wird erreicht, dass zunächst eine schnelle Hilfestellung für die Anwender im Vordergrund steht, während die tiefgründig technische Lösung erst im zweiten Schritt erfolgt.

Das ITIL-Modell sieht aber auch für die Fehlerbeseitigung dedizierte Aufgabenbereiche vor. So erfolgt nach der Problemidentifikation ein RFC (Request For Change – Änderungswunsch) an das Änderungsmanagement, welches die notwendigen Tätigkeiten zur Behebung des Problems initiiert. Die Protokollierung der Fehler und durchgeführten Änderungen kann damit gewährleistet werden.

Relevant für die Aufgabenteilung zwischen dem externen Dienstleister und der EDV des Krankenhauses ist darüber hinaus die Layer-Struktur von Services. So gliedern sich Betriebsleistungen für komplexe Services (Beispiel: KIS-Betrieb, Terminal-Server-Betrieb) in mehrere, funktional aufeinander aufbauende Ebenen:

Layer	Beschreibung
7	Geschäftsprozesse
6	Lösungen (Konzept, Planung, Design)
5	Applikationsbetrieb (Operating, Monitoring)
4	Datenbanken, Middleware
3	Betriebssysteme
2	Hardware/Plattform
1	Transport (LAN, WAN)

Grundsätzlich kann davon ausgegangen werden, dass je mehr die Aufgaben auf den unteren Ebenen angesiedelt werden können, desto besser eine Ausgliederung an einen externen Dienstleister erfolgen kann. Dies hat zur Ursache, dass Aufgaben der unteren Ebene weniger an die Spezifika des Hauses angepasst sind und der Anbieter daher Skalierungseffekte durch die Betreuung mehrerer Kunden nutzen kann. Umgekehrt gilt, dass Aufgaben der höheren Ebenen von der EDV des Krankenhauses wahrgenommen werden müssen, da eine Auslagerung mit hohen Kosten verbunden ist.

### 2.3. Ausschreibung von Dienstleistungen

Um zu einer Aufgabenteilung zwischen einem oder mehreren externen Dienstleistern und der EDV des Krankenhauses zu kommen, werden fest definierte Dienstleistungen in Form von Leistungsscheinen funktionell beschrieben und ausgeschrieben. Die Leistungsscheine beinhalten zum einen IT-Serviceleistungen auf Infrastrukturebene, wie beispielsweise der Aufbau und Betrieb des Netzwerks, zum anderen werden Querschnittsfunktionen ausgeschrieben, welche im Rahmen des IT-Service Managements notwendig sind für die Unterstützung der entsprechenden Arbeitsabläufe. Zu diesen zählen die Bereitstellung eines Trouble-Ticket Systems, einer Konfigurations-Datenbank, sowie eines Monitoringsystems.

## 3. Ergebnisse

Während die Organisation der EDV entsprechend ITSM in anderen Branchen auf Grund der Erfolge bereits weit verbreitet ist [1], ist in den Krankenhäusern diese Organisationsstruktur noch weitgehend unbekannt. Dies, obwohl die Strukturen zum

Teil komplexer als in anderen Branchen ist, beispielsweise dem Bankensektor. Ein Grund lässt sich in der Entwicklungshistorie der Krankenhaus-EDV finden: Während die Medizintechnik in den Häusern schon seit längerem eine etablierte Abteilung darstellt, wurde zumeist in den kleineren und mittleren Häusern erst in den letzten Jahren vermehrt in die EDV des Hauses investiert. Entsprechend stieg der Bedarf an IT-Mitarbeitern von einem „Mädchen für alles“ hin zu Abteilungen mit spezialisierten Mitarbeitern. Diese gewachsenen Strukturen gilt es mit Hilfe des ITIL-Modells in eine gerichtete Organisationsstruktur zu überführen. Erst mit dieser Einführung bewährter Best-Practice Methoden der Arbeitsorganisation lässt sich das mit der Durchführung von EAI-Projekten freigesetzte Potenzial eines stabilen, sicheren und kostengünstigen Betriebs realisieren.

Auch auf Seiten der Anbieter hat sich die Unterstützung einer nach ITIL organisierten EDV noch nicht durchgesetzt. So ergab die Ausschreibung, dass eine pauschale Bepreisung des Betriebs einzelner Dienstleistungen vielfach nicht möglich ist. Vielmehr arbeiten die Anbieter häufig nach dem Prinzip: „Betrieb der gesamten EDV“, oder „Bereitstellung von Ressourcen auf Stundenbasis“. Die Übernahme definierter Leistungsscheine mit vorgegebenen Service-Leveln ist hingegen vielfach noch nicht vorgesehen. So konnten nach der Anfrage bei 12 regionalen und überregionalen Anbietern von 3 Anbietern ein auf die Leistungsscheine passendes Angebot erarbeitet werden, 2 weitere Anbieter gaben nach längeren Verhandlungen Angebote ab, welche keine pauschale Bepreisung der Dienstleistungen vorsahen. Modellrechnungen mit den gewonnenen Ergebnissen deuten darauf hin, dass sich durch einen geschickten Servicemix entweder direkte Kosten reduzieren lassen, oder ein kosten-trächtiger Personalaufbau in der EDV-Abteilung des Krankenhauses vermieden werden kann.

#### **4. Zusammenfassung und Ausblick**

Mit Einführung eines vollständigen Fallpauschalensystems für die Abrechnung der Patientenbehandlung steigt bei den Krankenhäusern der Bedarf nach einem die IT-Systemlandschaft vereinheitlichenden Krankenhausinformationssystem (KIS). Abrechnungsrelevanten Behandlungsdaten können so direkt in den Behandlungsprozessen erfasst und dem administrativen System verfügbar gemacht werden. Auch die Steuerung der Behandlungsprozesse anhand im KIS hinterlegter klinischer Pfade wird mittelfristig umgesetzt werden.

Neben der Realisierung eines solchermaßen aufgestellten KIS ist es auch notwendig, eine entsprechende Organisationsstruktur für den Betrieb des KIS und der restlichen EDV des Krankenhauses zu erreichen. Diese Struktur muss zum einen den effizienten Einsatz verfügbarer Ressourcen gewährleisten, zum anderen müssen Kompetenzbereiche definiert und darüber hinaus benötigte Leistungen von externen Anbietern eingekauft werden. Dabei muss die Festlegung der von extern einzukaufenden Dienstleistungen auf dem Hintergrund erfolgen, dass hochspezialisiertes Wissen von der EDV-Abteilung eines Krankenhauses in der Regel nicht kostengünstig vorgehalten werden kann.

Als Lösung bietet sich eine Organisation entsprechend dem ITIL-Modell an. Hierbei wird unterschieden nach der Anwenderunterstützung in Form eines Service-

Desks, dem proaktiven Betrieb von Infrastruktur und Anwendungen, der Planung und Durchführung von Änderungen, sowie der Service Organisation. Für die Quantifizierung der Leistungen der IT-Abteilung wird das gesamte Aufgabenspektrum in Dienstleistungen aufgegliedert und deren Qualität in Form von Dienstleistungsvereinbarungen festgelegt. Werden Leistungen von externen Anbietern eingekauft, so übernehmen diese das mit einer festgelegten Dienstleistung verbundene Leistungsspektrum und die entsprechende Dienstleistungsvereinbarung. Eine kostentransparente Schnittstelle kann somit erreicht werden.

Während die Organisation der EDV-Abteilung nach ITIL weniger Schwierigkeiten bereitet, zeigt es sich, dass vielfach IT-Dienstleister nicht oder nur unzureichend auf die Bereitstellungen von IT-Dienstleistungen auf der Basis von vereinbarten Service-Leveln vorbereitet sind. Eine Aufteilung von Verantwortlichkeiten auf der Grundlage von Leistungsscheinen erfordert noch Klärungsbedarf. Die Anbieter sind an der Übernahme von IT-Dienstleistungen interessiert, so dass sich diese Form der Organisation auch in kleineren EDV-Abteilungen vermehrt durchsetzen wird.

## Literatur

1. Niemann H., Hasselbring W, Wendt T, Winter A, Meierhofer M.: Kopplungsstrategien für Anwendungssysteme im Krankenhaus. *Wirtschaftsinformatik* 44 (2002) 5, S. 425-434.
2. ITIL: Enterprise Application Integration. [www.itil.com](http://www.itil.com).
3. Warum gehört zu einem erfolgreichen EAI Projekt mehr als IT Know-How? Synaix Newsletter Ausgabe 02 / 2003, [www.synaix.de](http://www.synaix.de).
4. Winter AF, Winter A, Becker K, Bott O, Brigl B, Gräber S et al. (1999). Referenz-Modelle für die Unterstützung des Managements von Krankenhausinformationssystemen. *Informatik, Biometrie und Epidemiologie* 30(4), S. 173-189.
5. Macfarlane, I., Rudd, C.: *IT Service Management, Version 2*. itSMF Ltd Earley Reading RG6 7BH (GB) (1997)

