

# Applikationsübergreifendes Monitoring von Geschäftsprozessen

Thomas Bauer<sup>1</sup>, Ralph Bobrik<sup>2</sup>

<sup>1</sup> DaimlerChrysler Research and Technology, Abt. GR/EPD, Postfach 2360, D-89013 Ulm

<sup>2</sup> Universität Ulm, Abt. Datenbanken und Informationssysteme, D-89069 Ulm  
Thomas.TB.Bauer@DaimlerChrysler.com, Ralph.Bobrik@Uni-Ulm.de

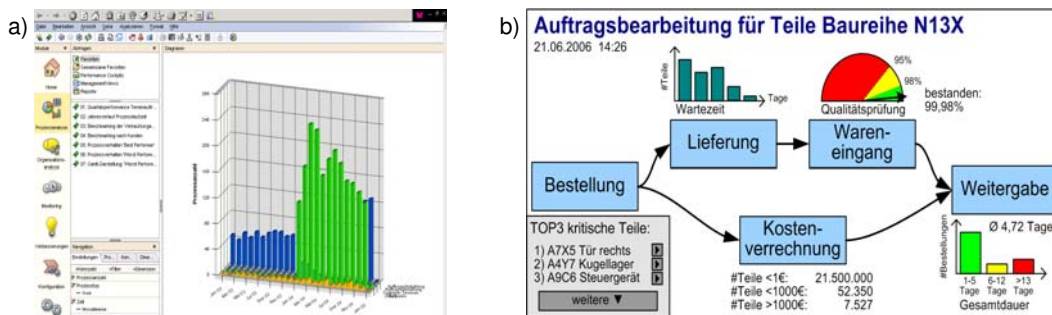
**Zusammenfassung.** Um die Bearbeitung von Geschäftsprozessen kontrollieren und optimieren zu können, ist ein Monitoring ihrer Ausführung erforderlich. Hierfür gibt es zwar leistungsfähige Werkzeuge, es ist aber aufwendig, diese an die Prozess-Ausführungskomponenten anzubinden. Deshalb wird untersucht, wie eine generische und wiederverwendbare Anbindung realisiert werden kann, falls die Applikation durch ein Workflow-Management-System (WfMS) gesteuert wird. Dies ermöglicht ein bereichs- und applikationsübergreifendes Monitoring mit geringem Aufwand für die einzelnen Anwendungsprojekte.

## 1 Einleitung

Applikationsintegration kann durch Kopplung von Anwendungen realisiert werden. Bei Großkonzernen mit bereichsübergreifenden Applikationen ist hierfür eine Vereinheitlichung der entsprechenden Schnittstellen hilfreich, um die Komplexität zu beherrschen [1]. In Umgebungen mit hunderten von Applikationen ist aber keine derart enge Kopplung zwischen all diesen Anwendungen erreichbar: Punkt-zu-Punkt-Verbindungen und sogar Medienbrüche sind auf längere Zeit unvermeidbar. Dennoch ist eine (betriebswirtschaftliche) Steuerung und Optimierung der entsprechenden Gesamt-Geschäftsprozesse notwendig. Dies wird durch das Monitoring der Prozessausführung ermöglicht, was aber IT-technisch geeignet unterstützt werden muss.

Heutzutage gibt es mächtige Tools zum Performance-Management, welche diese Funktionalität unterstützen, wie z.B. ARIS Process Performance Manager (PPM) [8] (siehe Abb. 1a) und WebSphere Business Monitor [6]. Ihr maximaler Nutzen entsteht durch die Integration in eine Prozessvisualisierung [2, 3], weil Kennzahlen und Diagramme in ihrem Prozesskontext dargestellt werden können (vgl. Abb. 1b). Ein generelles Problem solcher Performance-Manager ist allerdings, dass sie an diejenigen Applikationen angebunden werden müssen, welche die Runtime-Daten verwalten. Dies erfordert eine aufwendige Implementierung von Adaptern.

Performance-Management und Prozess-Visualisierung [2, 3, 4] sind in der wissenschaftlichen Literatur aufkommende Themen, deren Bedeutung zunehmend erkannt wird [5]. Die Anbindung von WfMS wäre prinzipiell einfach möglich, weil diese ohnehin über die relevanten Prozess- und Ausführungsinformationen verfügen. Deshalb wird in diesem Bericht untersucht, wie eine entsprechende Anbindung so reali-



**Abb. 1.** a) Process-Performance-Management und b) Integration in eine Prozessvisualisierung

siert werden kann, dass der Aufwand für die einzelnen Anwendungsprojekte reduziert wird. Zwar gibt es für bestimmte Paare von WfMS und Performance-Manager bereits (proprietäre) Integration (z.B. ARIS PPM mit Staffware, IBM WebSphere Monitor mit IBM Process Server), allerdings existieren hierfür bisher keine allgemeingültigen Architekturen oder Konzepte. Deshalb werden im Abschnitt 2 geeignete Ansätze zu Kopplung der Systemtypen untersucht. Abschnitt 3 skizziert geeignete standardisierte Schnittstellen und geht auf Umsetzungsmöglichkeiten mit kommerziellen Systemen ein, bevor Abschnitt 4 die Ergebnisse zusammenfasst.

## 2 Design-Varianten zur Kopplung der Systeme

Grundidee der nachfolgend vorgestellten Ansätze ist, einmalig zentral zu spezifizieren, welche Ereignisse und Daten beim Performance-Management gemessen werden.

**Ausbaustufe 1:** Im einfachsten Fall erfolgt diese Spezifikation wie in Abb. 2a dargestellt mit einem Administrationswerkzeug. Bei diesem kann es sich z.B. um einen Text- oder XML-Editor handeln, mit dem die entsprechenden Daten manuell spezifiziert werden. Die Informationen über die relevanten Messpunkte und -daten werden an den WfMS-Adapter übergeben. Dieser verwendet das Workflow-API des jeweiligen WfMS, um die entsprechenden Ereignisse zu erkennen und auf die benötigten Daten zuzugreifen. Da dieser Zugriff von den durch das Workflow-API angebotenen Funktionalitäten abhängt, ist der WfMS-Adapter produktspezifisch zu realisieren.

Auf den WfMS-Adapter selbst kann aber über ein vereinheitlichtes Performance-Management-API zugegriffen werden. Dies hat den Vorteil, dass über dieses API beliebige WfMS in das Performance-Management eingebunden werden können, nachdem für das WfMS (einmalig und prozessunabhängig) ein entsprechender WfMS-Adapter implementiert wurde. Der Performance-Manager (und sein zugehöriger Performance-Management-Adapter) müssen also nicht an unterschiedliche WfMS angepasst werden. Zusätzlich können aufgrund der einheitlichen Schnittstelle problemlos mehrere WfMS oder auch Legacy-Applications an denselben Performance-Manager angebunden werden, was ein applikationsübergreifendes Monitoring ermöglicht.<sup>1</sup> Analog kann ein WfMS-Adapter in Kombination mit unterschiedlichen Perfor-

<sup>1</sup> Die Identifikation zusammengehörender Prozessinstanzen findet wie immer beim Performance-Manager mittels eines Anwendungsdatums wie z.B. einer Auftragsnummer statt.

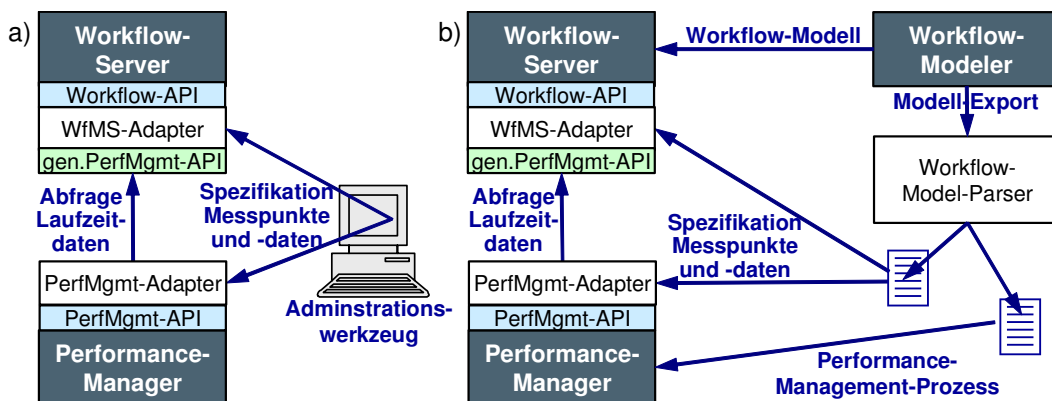


Abb. 2. Engineering-Ansatz für a) Ausbaustufe 1 und b) Ausbaustufe 2

mance-Management-Produkten verwendet werden, da alle Performance-Management-Adapter dasselbe API verwenden. Aufgabe eines solchen Adapters ist es, je nach Performance-Management-Produkt, Statusabfragen an das WfMS weiterzuleiten bzw. (z.B. periodisch) zu initiieren. Außerdem wandelt er die Ergebnisse in die vom jeweiligen Performance-Manager benötigte Form.

Die Ableitung von Performance-Kennzahlen und -Diagrammen (z.B. Gesamt-Prozesskosten) aus den Basisinformationen ist nur lokal für den Performance-Manager relevant. Deshalb werden diese dort mit den üblichen Methoden definiert, weshalb kein gesondertes (einheitliches) API und keine zusätzliche Spezifikationsprache verwendet werden.

**Ausbaustufe 2:** Ein weitergehender Lösungsansatz ist, die Spezifikation der relevanten Messpunkte und -daten nicht mit einem separaten Werkzeug zu erstellen, sondern hierfür das Modellierungswerkzeug des WfMS zu verwenden. Im Buildtime-Tool eines WfMS werden ohnehin das Workflow-Modell inkl. der Ein- und Ausgabedaten aller Aktivitäten definiert. Für jedes für das Monitoring relevante Modell wird nun für die Aktivitäten zusätzlich festgelegt, ob sie für das Performance-Management relevant sind. Außerdem wird angegeben, welche Daten beim Eintreten eines Ereignisses an das Performance-Management übermittelt werden sollen. Allerdings sind die Modellierungs-Tools heutiger WfMS hierfür typischerweise nicht erweiterbar. Dies führt dazu, dass gewisse Work-arounds notwendig werden, um in dem Tool die benötigten Informationen festlegen zu können. So können hierfür z.B. benutzerdefinierbare Attribute, reservierte Datenelemente oder Kommentarfelder verwendet werden. Diese Vorgehensweisen sind zwar etwas unschön, aber durchaus akzeptabel, da das im Modellierungs-Tool zur Verfügung stehende graphische Prozessmodell und die zentrale Informationsspezifikation zu einer deutlich erhöhten Übersichtlichkeit führen.

Das erstellte Workflow-Modell wird (wie üblich) an den Workflow-Server zur Prozesssteuerung übergeben (siehe Abb. 2b), da er die Ausführungskomponente für die Workflows darstellt. Außerdem wird das gesamte Prozessmodell inkl. der zusätzlich spezifizierten Daten exportiert. Hierfür bieten Modellierungswerkzeuge üblicherweise ein (leider häufig produktspezifisches) XML-Format an. Diese Prozessbeschreibung wird dann geparkt, um daraus den Input zur Steuerung des WfMS-Adapters abzuleiten. Es handelt sich hierbei um dieselbe Beschreibung, die bei der Ausbaustufe 1 manuell erstellt wird. Da sie aber automatisch abgeleitet wird, wird durch dieses Vorgehen ihre Konsistenz gewährleistet. Außerdem reduziert der Ansatz den Aufwand

für die Erstellung und auch Wartung dieser Spezifikation: Da die benötigten Informationen im Workflow-Modell enthalten sind, muss im Fall von dessen Änderung lediglich ein neuer Export und Parserlauf durchgeführt werden, um eine angepasste Spezifikation für die Kopplung zu erhalten.

Der Parser erzeugt, wie in Abb. 2b dargestellt, außer der bereits erwähnten Spezifikation noch eine weitere für den Performance-Manager. Diese enthält eine Beschreibung der Prozessstruktur. Diese ist nicht identisch mit der Prozessstruktur des Workflow-Modells, da nur bestimmte Aktivitäten und -übergänge für das Performance-Management und Prozessvisualisierungen relevant sind. Die Prozessstruktur ist also eine „verdichtete“ Darstellung des Prozesses, die aus ausgewählten Aktivitäten und entsprechend neu erzeugten Kanten besteht. Da diese Prozessstruktur (die sonst manuell im Performance-Manager definiert wird), automatisch aus dem eigentlichen Workflow-Modell abgeleitet wird, ist die Spezifikation fehlerfrei und kann mit deutlich geringerem Aufwand erstellt werden. Allerdings sind Aktivitäten, die von Legacy-Applications ausgeführt werden, nicht im Prozessmodell enthalten, so dass dieses noch (manuell) nachbearbeitet werden muss. Dies ist ebenfalls dann erforderlich, wenn mehrere Einzelprozesse des Workflow-Servers beim Performance-Management zu einem einzigen (übergreifenden) Geschäftsprozess zusammengeführt werden.

**Ausbaustufe 3:** Ein noch umfassenderer Ansatz ist, alle benötigte Information bei der Geschäftsprozess-Modellierung (z.B. in ARIS) festzulegen. Hierbei wird (im Gegensatz zur technischen Spezifikation bei der Workflow-Modellierung) eine deutlich stärker betriebswirtschaftlich geprägte Sicht modelliert, die sich deshalb besser zum Performance-Management eignet. Aus diesem Modell kann dann das Workflow-Modell abgeleitet werden. Wird das Geschäftsprozess-Modell (wie beim vorherigen Ansatz) um zusätzliche Information über relevante Messpunkte und -daten angereichert, so kann daraus automatisch der Input für den WfMS-Adapter berechnet werden.

Analog kann auch die vom Performance-Manager benötigte Information aus dem Geschäftsprozess-Modell abgeleitet werden. Hierzu werden an den Performance-Manager außer den Messpunkten und -daten auch noch Informationen über den „verdichteten“ Prozess übermittelt, so dass dort nur noch die benötigten Diagrammtypen definiert werden müssen. Aus Sicht des Performance-Managers verhält sich der Ansatz also ebenso wie Ausbaustufe 2. Er ermöglicht allerdings zusätzlich die Modellierung von Aktivitäten aus Legacy-Applications und die übergreifende Modellierung mehrerer Prozesse ggf. unterschiedlicher Workflow-Server. Damit kann an den Performance-Manager bereits der endgültige Gesamtprozess exportiert werden.

**Wertung:** Bei Ausbaustufe 1 erfordert die Erstellung der Steuerungsdatei einen hohen manuellen Aufwand. Ausbaustufe 3 ermöglicht eine absolut zentrale Informationsbereitstellung für alle beteiligten Komponenten. Allerdings ist das Verfahren nur einsetzbar, wenn das Workflow-Modell ohnehin aus dem Geschäftsprozess-Modell abgeleitet wird, was technisch und inhaltlich sehr schwierig ist und deshalb selten der Fall ist. Außerdem wird manchmal überhaupt keine von der Workflow-Definition getrennte Geschäftsprozess-Modellierung durchgeführt, so dass das Verfahren auch in diesen Fällen nicht anwendbar ist. Ausbaustufe 2 erscheint als am besten geeignet, da sie eine hohe Konsistenz bei geringem projektspezifischem Konfigurationsaufwand ermöglicht. Die mit dieser Lösung verbundenen Schwierigkeiten bei Aktivitäten aus Legacy-Applications und der Zusammenführung von systemübergreifenden Work-

flows sind leicht lösbar. Außerdem ist der Aufwand für die Erstellung der Basis-Infrastruktur deutlich geringer als bei der Ausbaustufe 3.

### **3 Realisierung der Schnittstellen und Abbildung auf Produkte**

Alle Ausbaustufen verwenden eine Beschreibung von Messpunkten und -daten für den WfMS-Adapter und den Performance-Management-Adapter, die deren Laufzeitverhalten steuert. Die hierfür benötigte Steuerungsinformation wurde detailliert untersucht. Aus Platzgründen kann ihr Aufbau im Folgenden lediglich skizziert werden:

Den betroffenen Geschäftsprozess und den zugehörigen Workflow-Typ.

Alle für das Performance-Management relevanten Prozessschritte und die zugehörigen Workflow-Aktivitäten.

Die vom Monitoring betroffenen Attribute und ihre Abbildung auf Workflow-Daten. Hierbei wird (für den WfMS-Adapter) zusätzlich spezifiziert, wie im WfMS auf diese Daten zugegriffen werden kann (z.B. Out-Container, prozessglobale Variable) bzw. wie ein Zugriff auf externe Daten möglich ist (z.B. JDBC). Zur Korrelation spezifiziert ein Flag, ob das Attribut vom Performance-Manager zur Identifikation der Prozessinstanz verwendet werden soll (d.h. Teil der ID ist).

Die Runtime-Schnittstelle zur Übermittlung von Ausführungsstatus und -daten vom WfMS-Adapter zum Performance-Management-Adapter wurde ebenfalls definiert. Mittels dieser kontaktiert der Performance-Management-Adapter einen WfMS-Adapter und spezifiziert den Zeitpunkt, ab dem Aktionen für ihn relevant sind (d.h. von wann die letzten Ergebnisse stammen, die er erhalten hat). Die Antwort identifiziert ausgeführte Prozessschritte und beschreibt zugehörige Daten mittels einer Liste von Attributnamen und -werten (Name-/Value-Pairs). Diese Liste muss mindestens alle diejenigen Attribute enthalten, über welche der Performance-Manager die Prozessinstanz identifiziert. Typischerweise enthält die Antwort auch stets den Bearbeiter des Prozessschrittes oder seine Organisationseinheit und den Zeitpunkt der Ausführung (evtl. Start- und Endezeit). Außerdem wird der Bearbeitungszustand der zugehörigen Aktivitäteninstanz übermittelt (z.B. Running, Completed). Hierbei ist es allerdings möglich, dass ein Performance-Manager nur die Beendigung von Aktivitäten verarbeiten kann und ein Eintrag ansonsten vom Performance-Management-Adapter entfernt wird. Analog können bei bestimmten Workflow-Servern oder Applikationen gewisse Ereignisse nicht erkannt werden (z.B. nicht der Start von Aktivitäten, sondern nur deren Beendigung). Es ist möglich, diese Schnittstellen auf ein XML-Austauschformat abzubilden. Je nach Zielumgebung kann z.B. aber auch eine Web-Service-Kommunikation mit einer direkten Parameterübergabe besser geeignet sein.

Um die Konzepte zu überprüfen, haben wir ihre Umsetzbarkeit mit kommerziellen Produkten analysiert: Bei dem WfMS MQ Workflow ermöglicht eine Datenbank-View (FMC.PROG\_ACT\_INST\_VIEW) [7] dem WfMS-Adapter die Ermittlung aller relevanter Zustandsübergänge, weil in dieser die Zeitstempel für das Eintreten aller relevanten Aktivitätenzustände aufgelistet werden. Diese View ermöglicht den Zugriff auf die wichtigsten Attribute per SQL, und das gezielt für diejenigen Ereignisse, die seit der letzten Abfrage eingetreten sind. Ebenfalls vorteilhaft an der Verwendung

einer Datenbankoperation ist, dass alle Ereignisse (effizient) mit einer einzigen Anfrage ermittelt werden können. Lediglich auf Anwendungsdaten und Mitarbeiterinformationen der Aktivitäten muss für jede ermittelte Aktivität einzeln per Workflow-API [7] zugegriffen werden. Das Workflow-Modell wird bei MQ Workflow mittels einer FDL-Datei von der Modellierungskomponente zum Workflow-Server transferiert. Des Weiteren ist die Anreicherung des Workflow-Modells um Zusatzinformation möglich, die ebenfalls in die FDL-Datei geschrieben wird. Damit sind bei der Ausbaustufe 2 alle für den Workflow-Model-Parser benötigte Informationen verfügbar.

Der ARIS PPM erhält die Information über den Ausführungszustand als sog. XML-Protokolldatei [8]. Es ist für den Performance-Management-Adapter leicht möglich, die mittels der erwähnten Runtime-Schnittstelle empfangenen Daten in dieses Format zu transformieren. Die Struktur des Prozesses wird dem PPM im Ereignisformat [8] übermittelt. Dieses Format kann vom Workflow-Model-Parser aus der FDL-Prozessdefinition erzeugt werden. Des Weiteren ist es im Falle eines Workflow-übergreifenden Monitorings sehr einfach, mehrere generierte Abläufe zu verketten oder Aktivitäten aus Legacy-Applications einzubinden. Dies liegt daran, dass das Ereignisformat nur Einzelfragmente mit Startereignis-Funktion-Endereignis spezifiziert, so dass lediglich Ereignisnamen verändert oder zusätzliche Fragmente definiert werden müssen. Die gewünschten Kennzahlen und Diagramme lassen sich mittels der in Abb. 1a dargestellten Oberfläche des PPM leicht definieren.

## 4 Zusammenfassung

Für das Berichtswesen und zur Prozessoptimierung ist ein Monitoring von Applikationen erforderlich. Wir haben aufgezeigt, wie hierzu WfMS über eine standardisierte Schnittstelle und Methodik effizient an ein Performance-Management angebunden werden können. Eine derartige Vereinheitlichung ist in großen Anwendungsverbänden unerlässlich, um den entstehenden Aufwand bewältigen zu können. Entsprechende Schnittstellen werden zudem benötigt, um einer benutzerspezifischen Prozessvisualisierung [2, 3] den Zugriff auf die Ausführungsdaten der Prozesse zu ermöglichen.

### Literatur

1. T. Bauer: Integration von prozessorientierten Anwendungen. In: Proc. Workshop on Enterprise Application Integration, Marburg. (2005) 66-73
2. 1. Int. Workshop on Business Process Monitoring & Performance Management. In: Proc. 16th Int. Workshop on Database and Expert Systems Applications. (2005)
3. R. Bobrik, T. Bauer und M. Reichert: Proviado - Personalized and Configurable Visualizations of Business Processes. In: Proc. 8th Int. Conf. on Electronic Commerce and Web Technologies, Krakau. (2006) 61-71
4. R. Bobrik, M. Reichert und T. Bauer: Requirements for the Visualization of System-Spanning Business Processes. In: [2]. (2005) 948-954
5. F. Casati: Industry Trends in Business Process Management: Getting Ready for Prime Time. In: [2]. (2005) 903-907
6. IBM: WebSphere Business Monitor 6.0, Data Sheet. (2005)
7. IBM: WebSphere MQ Workflow Version 3.6 - Programming Guide, 11. Auflage. (2005)
8. IDS Scheer: ARIS Process Performance Manager - Techn. Referenz Datenimport. (2005)