

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-

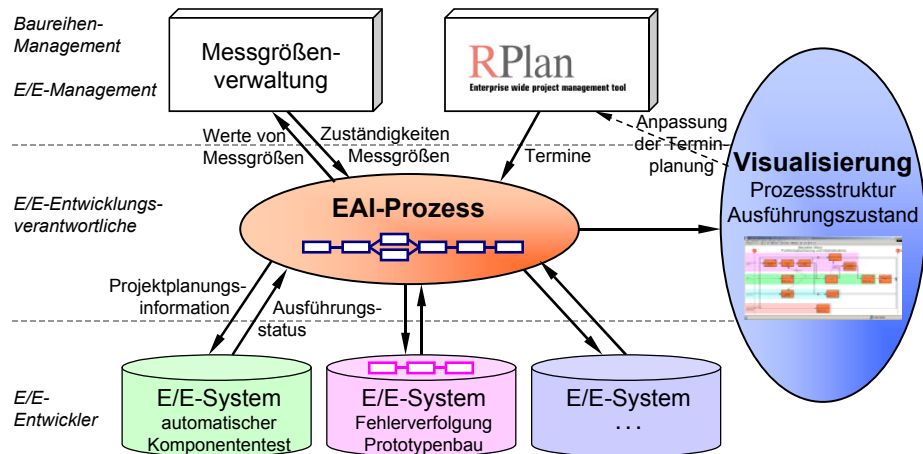


Abb. 1. Verwendung und Datenaustausch zwischen den integrierten Anwendungssystemen

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-

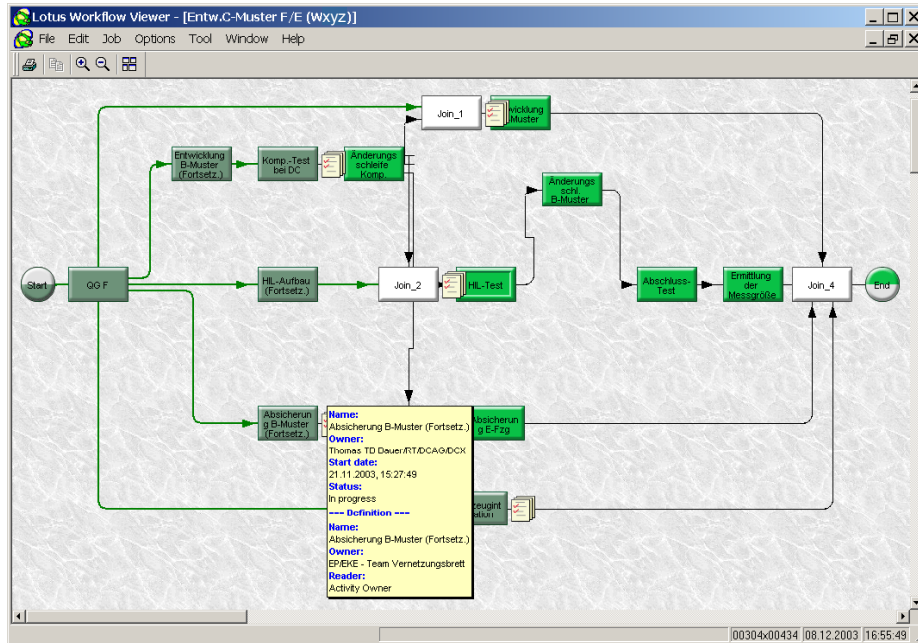


Abb. 2. Visualisierung einer Prozessinstanz in Lotus Workflow

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.

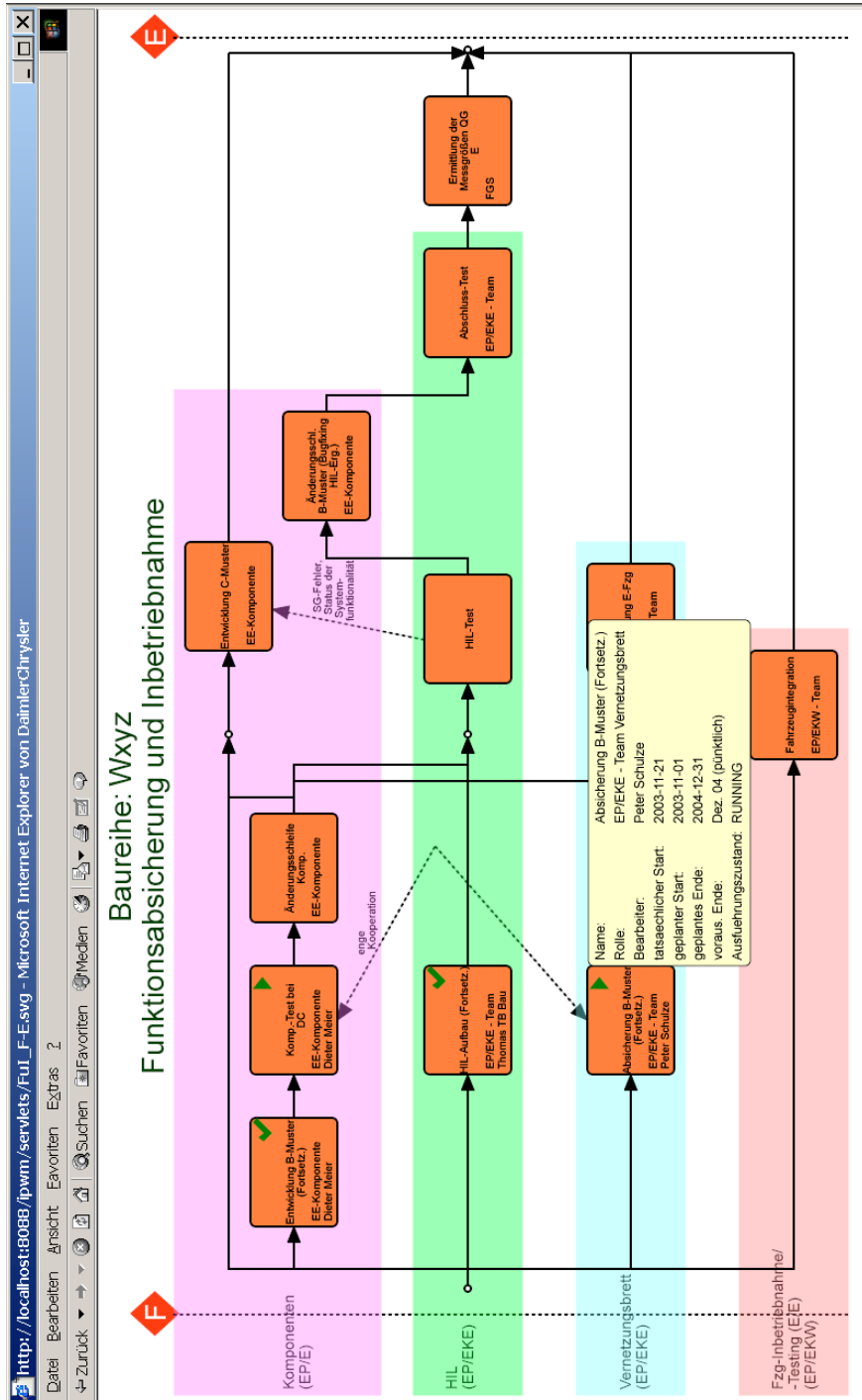


Abb. 3. Benutzergerechte Visualisierung einer Workflow-Instanz

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)