

Anpassungsfähige Workflows zur Unterstützung unstrukturierter Vorgänge

Reiner Siebert

Universität Stuttgart,
Institut für Parallele und Verteilte Höchstleistungsrechner (IPVR)
Breitwiesenstr. 20-22, D-70565 Stuttgart
e-Mail: siebert@informatik.uni-stuttgart.de
<http://www.informatik.uni-stuttgart.de/ipvr/as/personen/siebert.html>

Zusammenfassung:

Um Workflow-Management-Systeme (WFMS) flexibel und vielseitig einsetzen zu können, müssen auch Vorgangsteile unterstützt werden, die nicht vollständig determinierbar bzw. unstrukturiert sind. Nach einer kurzen Motivation und Einführung in die Umgebung des Projektes PoliFlow wird eine Konzeption vorgestellt, die eine einheitliche Vorgangssteuerung von strukturierten und unstrukturierten Prozessen ermöglicht.

Anschließend wird besonders auf manuelle Anpassungen einzelner Workflow-Instanzen eingegangen. Dabei wird ein Modell diskutiert, mit dem diese Anpassungen hinreichend kontrolliert werden können, um unerwünschte oder fehlerhafte Eingriffe zu vermeiden. Eine wesentliche Rolle spielen hierbei auch benutzerdefinierte Anpassungsrechte, die weiter vertieft werden. Der Beitrag schließt mit einem Überblick über den aktuellen Stand und weitere Zielsetzungen der Arbeiten.

Einleitung

In der Vergangenheit wurden WFMS lediglich zur effizienten Steuerung häufig auszuführender, stark strukturierter Vorgänge eingesetzt. Die Unterstützung unstrukturierter Prozesse hingegen erfolgte mit Hilfe typischer CSCW-Werkzeuge. In der Zwischenzeit wurden jedoch die Einschränkungen dieser Trennung erkannt und es ist ein starker Trend in die Richtung eines einheitlichen, unternehmensweiten Prozeßmanagements zu erkennen [8].

Vorgangssteuerung in einer weit verteilten Anwendungsumgebung innerhalb der öffentlichen Verwaltung ist auch die Aufgabenstellung im Verbundprojekt PoliFlow (gefördert vom BMBF). Durch die Vielfalt der Aufgaben in der öffentlichen Verwaltung muß ein geeignetes System in der Lage sein, unterschiedlichste Vorgangstypen zu unterstützen [6]. Neben stark strukturierten Vorgängen, bei denen alle zur Ausführung erforderlichen Details bereits während der Modellierung bekannt sind, existieren auch sehr unstrukturierte Vorgänge, bei denen wichtige Kontrollinformationen erst während der Durchführung ermittelt werden können. Aus diesem Grund wird der Forschungsschwerpunkt der *Adaptiven Workflow-Systeme (AWS)* im Projekt bearbeitet.

Neben der Vielfalt der Vorgangstypen zeichnet sich das Anwendungsfeld jedoch auch durch eine ausgesprochen heterogene Systemlandschaft und unterschiedlichste Anwendergruppen aus, die gemeinsam mit dem System arbeiten sollen. Diesen Faktoren wur-

den bei der Erstellung des Stuttgarter Workflow- und Telekooperationssystems (SWATS) berücksichtigt [3], das auf innovativen Technologien basiert (wie z.B. Intra-/Internet, Java, C++, CORBA etc.).

Konzeption anpassungsfähiger WFMS

Im Forschungsbereich AWS wurde ein Referenzmodell für anpassungsfähige Workflows entworfen. Durch flexible Konzepte auf verschiedenen Ebenen können dabei bestehende WFMS so erweitert werden, daß auch eine hinreichende Unterstützung von unstrukturierten Prozessen gewährleistet werden kann [5]. Neben der Übertragbarkeit, war eine weitere Anforderung, daß prinzipiell alle Workflow-Anwender, falls erforderlich, Anpassungen durchführen können sollten.

Das Konzept basiert einerseits auf der Vermeidung bzw. Automatisierung von Anpassungen, was durch die Verwendung einer flexiblen Workflow-Engine mit einem mächtigen Workflow-Modell erreicht wird (z.B. durch die Verwendung von Ereignissen bzw. deskriptiven und komplexen Kontrollkonstrukten).

Andererseits sind jedoch auch manuelle Anpassungen von Workflow-Instanzen während der Ausführung möglich, die durch verschiedene Kontrollmechanismen eingeschränkt werden können. Neben systeminternen Konsistenzprüfungen zur Sicherung der syntaktischen Korrektheit, werden auch benutzermodellierete Anpassungsrechte und Integritätsbedingungen kontrolliert, um die semantische Korrektheit der Workflow-Instanzen sicherstellen zu können.

Damit prinzipiell beliebige Workflow-Partizipanten, die über kein spezielles Expertenwissen verfügen, Anpassungen vornehmen können, sind besonders übersichtliche und intuitive Benutzerschnittstellen erforderlich. Manuelle Anpassungen können daher über einen graphischen Workflow-Editor durchgeführt werden, wobei den Anwendern zusätzlich Bibliotheken bereitgestellt werden, die neben fertigen Aktivitäten auch vordefinierte Standardanpassungen beinhalten.

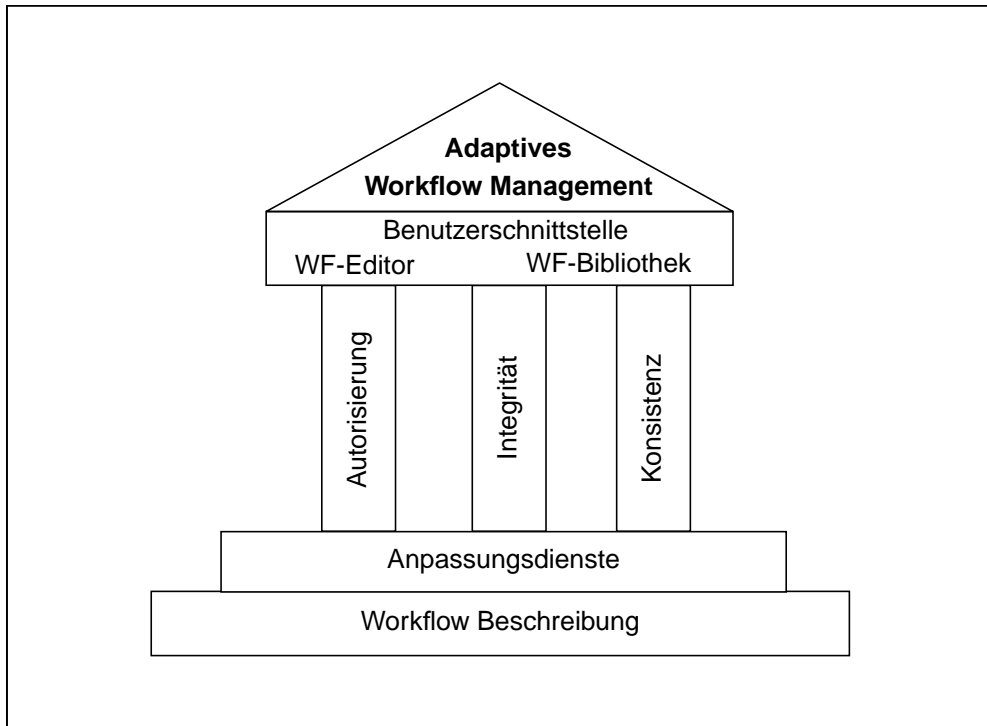


Abbildung 1: Lösungsansatz für adaptive Workflows

Neben der Benutzbarkeit durch beliebige Beteiligte, unterscheidet sich diese Konzeption von anderen Ansätzen [1,2,4,7] zum einen dadurch, daß erforderliche Anpassungen nicht in Aktivitätsimplementierungen versteckt sind, sondern explizit vorgesehen werden. Zum anderen beinhaltet sie neben automatisierten Anpassungen eben auch kontrollierte manuelle Änderungen, die wegen unvorhersehbarer externer Ereignisse nie auszuschließen sind.

Manuelle Workflow-Modifikationen

Wie bereits erwähnt, können immer Situationen eintreten in denen manuelle Anpassungen notwendig werden, falls z.B. unvorhergesehene Ereignisse eintreten, die Änderung einzelner Workflow-Instanzen erfordern. Darüber hinaus können manuelle Anpassungen auch erwünscht sein (z.B. bei Entscheidungspunkten oder bei individueller Aufgabenverteilung). Aus diesem Grund wird ein Satz von definierten Änderungen, die auf der Instanzbeschreibungen der Engine aufsetzen, durch einen Anpassungsdienst angeboten. Damit kann praktisch ein Reengineering einzelner Workflows durchgeführt werden. Diese Möglichkeit kann aber auch bewußt eingesetzt werden, um z.B. Verantwortungsbereiche explizit zu modellieren oder die Modellierung eines komplexen Workflows an einer bestimmte Stelle abubrechen, auch wenn noch nicht alle Ausnahmen abgedeckt sind. Der Vorgang wird dann bei einem typischen Verlauf ganz normal behandelt, in Ausnahmefällen greifen die Bearbeiter jedoch gezielt mit Anpassungen ein.

Bei streng begrenzten Anpassungen wie der Zuweisung eines speziellen Aktors oder dem Einfügen einer unproblematischen Zusatzaktivität ist dies zwar noch überschaubar, bei komplexeren Änderungen können die Auswirkungen von den Bearbeitern jedoch kaum mehr überblickt werden.

Darüber hinaus kann dieselbe Anpassung in einem Fall erwünscht und ausgesprochen hilfreich sein, in einem anderen Fall könnte sie jedoch fatale Folgen haben. Ist z.B. eine Folgeaktivität überflüssig geworden, so ist es wünschenswert, wenn diese gelöscht werden kann. Eine Folgeaktivität, die hingegen Kontrollzwecke verfolgt, sollte wiederum nicht so einfach übergangen werden können.

Aus diesen Beispielen wird bereits deutlich, daß Anpassungen sehr schwierig kalkulierbar sein können, weshalb zusätzliche Kontrollmechanismen unbedingt erforderlich sind. Durch die Kontrollen soll vermieden werden, daß

- Benutzer eigenständig unerwünschte oder unerlaubte Anpassungen vornehmen,
- durch manuelle Anpassungen die syntaktische Korrektheit der Workflow-Beschreibung gefährdet wird oder daß
- unreflektierte Anpassungen zu Auswirkungen führen, welche die semantische Korrektheit des Workflows beeinträchtigen.

In WFMS, die keine Änderungen an der Workflow-Instanzbeschreibung vorsehen, sind diese Prüfungen zur Laufzeit nicht erforderlich, da die Zustandsübergänge der Instanzen ausschließlich durch die Workflow-Definition und die Engine selbst bestimmt werden.

Da Konsistenz- und Integritätsprüfungen von Workflow-Definitionen, selbst während der Modellierungsphase nicht-triviale Aufgaben darstellen, müssen Mechanismen bereitgestellt werden, um diese zur Laufzeit effizient zu gestalten. Dafür ist es notwendig, daß potentielle Auswirkungen von Anpassungen berechenbar sind und im Falle einer Anpassung die Bedingungen aller betroffenen Objekte wieder ausgewertet werden. Falls diese nicht erfüllt werden können, ist die Anpassung in der Regel abzuweisen.

Weiterhin wurde deutlich, daß bestimmte Anpassungen unabhängig von Konsistenz- und Integritätsprüfungen, zugelassen, andere jedoch verhindert werden müssen. Welcher Anwender welche Rechte besitzt, wird in einem neuen Aspekt des Workflow-Modells definiert. Im folgenden wird auf die Modellierung dieser Anpassungsrechte näher eingegangen.

Autorisierung manueller Anpassungen

Durch die Definition von Anpassungsrechten soll sichergestellt werden, daß genau die Anpassungen vorgenommen werden dürfen, die erwünscht sind und daß nicht autorisierte Anpassungsversuche scheitern. Diese Problemstellung kann wegen der Vielfalt der Anpassungen und der unterschiedlichen Eigenschaften der Prozesse nicht durch einen einfachen und starren Ansatz gelöst werden.

Ob und unter welchen Umständen eine Anpassung zulässig ist hängt von verschiedenen Faktoren ab (vgl. Abbildung 2) und muß daher für jeden Workflow spezifisch geregelt werden. Dabei muß geklärt werden, welche Personen unter welchen Umständen (Zustand der Workflow-Instanz) welche Anpassungen auf einem Workflow durchführen dürfen. Darüber hinaus können verschiedene Maßnahmen (z.B. Benachrichtigung weiterer betroffener Workflow-Partizipanten) an die Durchführung von Anpassungen gebunden werden. Ein Anpassungsrecht kann also durch einen Quadrupel (Anpassung, Aktor, Zustand, Maßnahme) definiert werden.

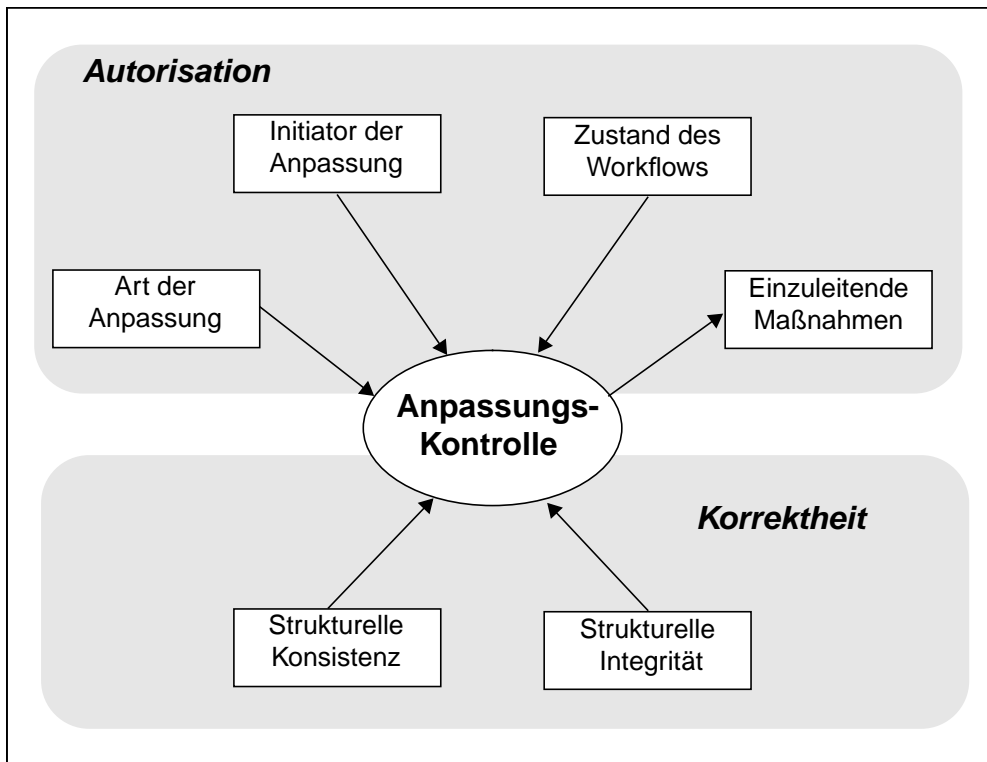


Abbildung 2: Einflussfaktoren einer Anpassungskontrolle

Mit diesem einfachen Modell könnten zwar flexible Rechte definiert werden, die Modellierung würde jedoch so aufwendig werden, daß sie kaum anwendbar wäre. Um die Definition zu vereinfachen ist das Modell wie folgt zu erweitern:

- Durch eine *Gruppierung* innerhalb der Faktoren können die Rechte global für bestimmte Anpassungstypen, Workflow-Typen und Benutzergruppen definiert werden.
- Durch die Erweiterung des Modells um höher priorisierte *Anpassungsverbote* können einzelne Rechte aus einer globalen Definition wieder ausgeschlossen werden.
- Durch die Einführung von *Merkmale*, wird eine zusätzliche Abstraktionsebene eingeführt, die es ermöglicht, festgelegte Standarddefinitionen von Anpassungsrechten bereits im System zu verankern (Default-Definition).

Mit Hilfe dieser Gruppierungen läßt sich die Definition der Anpassungsaspekte deutlich vereinfachen. Der Erfolg hängt jedoch auch von der Qualität der Typisierung und von der Eignung der Merkmale für ein bestimmtes Anwendungsgebiet ab.

Bei einer Analyse der Anpassungen im Anwendungsfeld des Projektes PoliFlow hat sich gezeigt, daß relevante Gemeinsamkeiten von Workflows (bzw. Anpassungen) existieren, die zur Typisierung herangezogen wurden. Dabei mußten vor allem die Gründe betrachtet werden, weshalb Anpassungen in Einzelfällen erwünscht oder unerwünscht sind. Als relevante Kriterien haben sich hier u.a. ergeben:

- Temporale Aspekte (z.B. verzögerndes vs. nicht verzögernde Anpassungen)
- Sicherheitsaspekte (z.B. Delegation von Aufgaben an bereits beteiligte vs. noch nicht beteiligte Personen)
- Schwache Strukturierung (freiere Anpassungsrechte z.B. in kooperativen Subworkflows)
- Starke Strukturierung (z.B. Anpassungen von normalen Bearbeitern in der Regel nicht erlaubt)

Stand der Arbeit und Ausblick

Die konzeptuellen Arbeiten in den Bereichen des Anpassungsdienstes und der Anpassungsrechte sind so gut wie abgeschlossen; mit der Erweiterung des Workflow-Modells um spezielle Anpassungsaspekte und deskriptive Konstrukte wurde bereits begonnen. Der Schwerpunkt der zukünftigen Arbeit wird in der Formalisierung der Konsistenz- und Integritätsbedingungen liegen.

Ein erster Prototyp wurde entwickelt, der einige Anpassungen der verhaltensorientierten Aspekte mit Hilfe eines graphischen Workflow-Editors unterstützt. Neben zusätzlichen Anpassungen werden hier Erweiterungen wie z.B. die Anpassungskontrolle eingebracht. Durch die Umstellung auf Java JDK 1.1 ist allerdings eine Reimplementierung erforderlich. Darüber hinaus müssen die Komponenten noch in die bestehende Infrastruktur von SWATS integriert werden.

Durch den gewählten Ansatz können nicht alle Schwierigkeiten bzgl. unstrukturierter Workflows gelöst werden. Vielmehr ist es die Zielsetzung, eine Infrastruktur zu schaffen, mit der vielseitige Anpassungen auf eine definierte Art und Weise unterstützt werden können. Ein erster kompletter Prototyp soll auf der CeBIT'97 im Rahmen des Projektes PoliFlow präsentiert werden.

Literatur und weitere Informationen

- [1]Amberg Michael: Modelling Adaptive Workflows in Distributed Environments. In Workshop on Adaptive Workflow, Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM). Basel, October 1996.
- [2]Jablonski S., Stein K., Teschke M.: Experiences in Workflow Management for Scientific Computing. In 8th International Workshop on Database and Expert Systems Applications (DEXA). France, 1997.
- [3]PoliFlow-Infonet: <http://www.informatik.uni-stuttgart.de/ipvr/as/projekte/poliflow/poliflow.html>
- [4]Reichert M., Dadam P.: A Framework for Dynamic Changes in Workflow Management Systems. In 8th International Workshop on Database and Expert Systems Applications (DEXA). France, 1997.
- [5]Siebert Reiner: Adaptive Workflow for the German Public Administration. In Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM), Workshop on Adaptive Workflow. Basel, 1996.
- [6]Siebert R., Kindler T., Soyez, T.: Integrated Workflow and Telecooperation Support for the German Government. In Proc. of the Symposium on Applied Computing (SAC). San Jose, 1997.
- [7]Wainer J., Weske M., Vossen G., Medeiros C.: Scientific Workflow Systems. In NSF Workshop on Workflow and Process Automation in Information Systems. Athens-Georgia, May 1996.
- [8]Wolf M., Reimer U.: Proc. of the First Int. Conf. on Practical Aspects of Knowledge Management (PAKM). Basel, October 1996.