

Workflow Management Systems: Formal Foundation, Conceptual Design, Implementation Aspects

— Habilitationsschrift —

vorgelegt am Fachbereich für Mathematik und Informatik
der Westfälischen Wilhelms-Universität Münster

von

Dr. rer. nat Mathias Weske

Summary

This thesis deals with the formal foundation, conceptual design, and prototypical implementation of workflow management systems. Based on properties of information systems applications, research issues in workflow management and requirements of suitable workflow support are characterized. In particular, the need for flexibility and dynamic adaptations is identified, i.e., the ability of workflow management systems to adapt running workflow instances to changes in the environment of application processes. Re-use of workflow schemata, the integration of application objects as well as high scalability and availability are characterized as additional important properties of workflow management systems.

As a formal foundation, a workflow language based on directed graphs is introduced, using a mathematical formalism. In this formalism, workflow schemas and workflow instances are described. Specific properties of workflow schemas like the prefix-closed property with respect to workflow schema consistency are formalized. Workflow application development processes are characterized as specific software processes, and a workflow design methodology is proposed.

The conceptual design of workflow management systems is studied in some detail. After a brief characterization of the conceptual design and the architecture of a workflow management system, the potentials for improvements are identified. Goals for the conceptual design and prototypical implementation of a novel workflow management system are derived. Since a mathematical formalization is not sufficient as a basis for the conceptual design of a workflow management system, object-oriented design methods are used. Workflow schemas, workflow instances, and related artifacts are modeled as objects. Since the resulting class diagram specifies the structure of workflow-relevant objects, it amounts to a workflow meta schema. The workflow meta schema is designed to support dynamic adaptations of running workflow instances by changing the relationship of a workflow instance object to a workflow schema object. To control dynamic adaptations properly, correctness criteria using the mathematical formalization are introduced.

Since workflow applications are typically performed in heterogeneous and distributed environments, a distributed object-oriented middleware is used as an implementation platform. Due to the object-oriented approach on the conceptual level, the system design and the implementation platform match nicely. A set of fundamental services are designed and implemented. The usage of the system is illustrated by a sample workflow application, which also introduces a workflow client application. The workflow client application can be configured to the requirements of different groups of persons involved in workflow applications.

Zusammenfassung

Diese Habilitationsschrift beschäftigt sich mit den formalen Grundlagen, dem konzeptionellen Design und der prototypischen Implementierung von Workflow-Management-Systemen. Es werden wissenschaftliche Fragestellungen identifiziert und Anforderungen an Workflow-Management-Systeme formuliert, die sich aus den Anforderungen moderner Anwendungen ableiten. Flexibilität wird als eine der zentralen Eigenschaften formuliert, insbesondere die dynamische Adaptierbarkeit laufender Workflow-Instanzen an neue Workflow-Schemata. Wiederverwendung von Workflow-Schemata und Anwendungsobjekten sowie hohe Skalierbarkeit und Verfügbarkeit werden als weitere zentrale Anforderungen an Workflow-Management-Systeme formuliert.

Als formale Grundlage für das konzeptionelle Design von Workflow-Management-Systemen wird eine auf gerichteten Graphen basierende Workflow-Sprache eingeführt und mathematisch formal beschrieben. In dem entwickelten Formalismus können spezielle Eigenschaften von Workflow-Schemata, etwa die Präfix-Abgeschlossenheit von Workflow-Schemata bezüglich Konsistenzigenschaften, gezeigt werden. Die Eigenschaften von Prozessen zur Entwicklung von Workflow-Anwendungen als spezielle Software-Entwicklungsprozesse werden diskutiert, und eine Methodologie zur Entwicklung von Workflow-Anwendungen wird entwickelt.

Der konzeptionelle Entwurf von Workflow-Management-Systemen wird ausführlich untersucht. Nach einer kurzen Charakterisierung von Entwurf und Architektur eines Workflow-Management-Systems werden Verbesserungsvorschläge formuliert. Ziele für den konzeptionellen Entwurf und die prototypische Implementierung eines neuen Workflow-Management-Systems werden hergeleitet. Weil die mathematische Formalisierung für den konzeptionellen Entwurf nicht hinreichend ist, werden objekt-orientierte Methoden verwendet. Workflow-Schemata und Workflow-Instanzen werden als Objekte modelliert. Das resultierende Klassendiagramm beschreibt die Struktur workflow-relevanter Objekte und stellt daher ein Workflow-Meta-Schema dar. Das Workflow-Meta-Schema erlaubt dynamische Adaptionen laufender Workflow-Instanzen dadurch, dass Beziehungen zwischen Workflow-Schema- und Workflow-Instanzobjekten zur Laufzeit verändert werden können. Korrektheitseigenschaften für dynamische Adaptionen werden auf der Basis des mathematischen Formalismus definiert. Basierend auf gültigen Abbildungen (valid mappings) kann das Workflow-Management-System entscheiden, ob eine Workflow-Instanz an ein gegebenes Workflow-Schema angepasst werden kann.

Weil Workflow-Anwendungen typischerweise in verteilten, heterogenen Umgebungen ausgeführt werden, wird als Implementierungsplattform eine verteilte, objekt-orientierte Middleware eingesetzt. Konzeptioneller Entwurf und Implementierungsplattform passen somit gut zueinander. Basisdienste werden entworfen und implementiert. Die Verwendung des Systems wird durch einen Beispiel-Workflow illustriert. Dabei wird auch eine Workflow-Client-Anwendung als Benutzerschnittstelle des Systems vorgestellt. Diese Anwendung kann unter Beachtung der funktionalen Anforderungen unterschiedlicher Benutzergruppen konfiguriert werden.

Gutachter

Prof. Dr. Gottfried Vossen, Westfälische Wilhelms-Universität Münster

Prof. Dennis Shasha, Ph.D., New York University

Prof. Dr. Gerhard Weikum, Universität des Saarlandes

Habilitation: 31. Mai 2000