Business Process Technology

Dirk Draheim

Business Process Technology

A Unified View on Business Processes, Workflows and Enterprise Applications

Springer

Dr. Dirk Draheim
Universität Innsbruck
Zentraler Informatikdienst (ZID)
Technikerstr. 23
6020 Innsbruck
Austria
draheim@acm.org

ISBN 978-3-642-01587-8 e-ISBN 978-3-642-01588-5
DOI 10.1007/978-3-642-01588-5
Springer Heidelberg Dordrecht London New York

Library of Congress Control Number: 2010932874

Cover design: KuenkelLopka GmbH

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

In the last decade there has been an explosion of interest in the modeling and automation of business processes, and competence in this area is seen as increasingly critical to business competitiveness and stability. However, this has lead to a parallel explosion in solution approaches and technologies leading to a state-of-the-art that is highly disjointed and confused. In particular, there is a mismatch between business process modeling technologies on the one hand, which focus on allowing domain experts to describe business processes in a graphical, easy-to-use way, and workflow engines on the other hand which focus on automating the enactment of business processes in association with human users. Not only is there little consensus on concepts and terminology, there is also little connection between commercial solutions and established computer science theory. This is where Dirk Draheim's book makes its contribution. First, it clarifies the conceptual differences and similarities between the many different business process technologies available today and lays the foundation for a unified understanding of the field. Second, it explores the relationship between these technologies and traditional principles of computer science such as structured programming. And third, it lays out a vision for the future of business process technology and its optimal use in business process improvement and enterprise systems development.

Most books on business process technology either take a very broad but high-level view of the challenges and solutions in this area or provide a very detailed but narrow view of a specific issue or technology. It is rare to find a book that manages to do both. Dirk Draheim combines his experience with the wide-range of practical technologies currently used to automate business processes with his deep understanding of computing science formalisms to show how the former can be given a stronger theoretical foundation. Finally the best part of the book is saved until the end. In the final chapter Dirk Draheim proposes "Typed Workflow Charts" as a new formalism for modeling and automating business processes. This represents a genuinely innovative step forward which is likely to have a big impact on the way business processes are specified and automated in the future.

Mannheim, July 2010 *Colin Atkinson*

Author's Preface

Is it possible to specify business processes in a technology-independent and executable manner? That is the question this book addresses. There are different communities addressing business processes each with different objectives, tools and terminology – business process reengineering, business process modeling, task modeling, business process management, workflow management. We seek for a unified understanding of the phenomena addressed by these communities. There is a huge potential for automation in today's Enterprises. An integrated platform for specifying and controlling processes in an enterprise would be an enabling technology to use this potential. However, there are severe challenges that must be overcome before such a platform can be designed. First, there are structural frictions in today's business process modeling and today's business process implementations, i.e., lack of operational semantics and lack of a canonical implementation. Second, current business process management (BPM) and workflow technologies are not fully integrated with the application programsthat implement the dialogues of an enterprise application.

Business process models do not have a precise operational semantics in the sense of a fixed set of rules that describe the state changes in the system under consideration. There is no canonical mapping between the activities of the business processes and the dialogues that support these activities. The workflow paradigm in its current form does not really help in this situation. Up to now, workflow technology is only really convincing in the field of document management. Current business process execution and management technologies arose as enterprise application integration technologies and they are still used in this manner. However, workflow technology is not yet a proven concept as a general enterprise resource management technology.

Today's BPM technology is successfully used in enterprise application projects in the following sense. As a first step the system analyst identifies the rules behind the interplay of existing enterprise applications. These rules are then automated by a BPM product. Today's BPM technology controls workflow states. However, it does not control the dialogues that bridge the workflow states – the dialogue states are not seen by BPM technology. This means,

most importantly, that the dialogues are also not amenable to advanced BPM tools and techniques like business process simulation and business process monitoring.

Furthermore, if BPM technology is used to build a workflow-intensive system from scratch it is not obvious any more how to design the human-computer interaction. The problem is to fix the right granularity of workflow states versus dialogue states. Despite some heuristics a systematic treatment of this question is still missing. We follow a different, more direct approach: workflow states and dialogue states are unified so that the aforementioned problem simply does not appear any more. This text aims at characterizing and mitigating the mentioned gaps. We target a seamless specification of workflows and dialogues.

Objectives of the Book

We analyze the existing gap between business process modeling, which is a system analysis activity, and business process automation, which is related to system design. We also analyze the gaps and tension between current classes of business process technology, i.e., business process modeling tools, workflow definition, and integrated development environments. We claim that an analysis of the aforementioned gaps and tension is necessary before an integrated business process management platform can be designed. These are some of the discussions, questions, results and contributions of the book:

- We explain that business process management lifecycle models should be understood as pools of systematic activities and argue that they can hardly be interpreted as strictly staged models in Sect. 2.4.1.
- We propose a new model of IT ownership which cleanly separates foreseeable total costs of ownership and assessable total benefit of ownership in Sect. 2.6.4.
- We introduce a spiral quality management system model in Sect. 2.7 which is reductionist in terms of organizational functions but sophisticated in terms of interfaces between organizational functions.
- We identify three distinguishable aspects of component technology in Sect. 3.3, i.e., the sub industry aspect, the infrastructure aspect and the large system construction aspect.
- We explain why today's emerging CSCW tools should be exploited in business continuity management in Sect. 3.4.
- We propose the integration of business processes, production processes and business intelligence by the means of data warehousing technology in Sects. 3.6.2.
- We distinguish between a global view on workflows, which is the view of workflow supervisory, and a local view on workflows, which is the view of the single workers involved in workflow executions, in Sect. 4.1. It turns out that this distinction helps in the understanding of quality of design

of business process specifications and also helps in understanding the gap between business process modeling and business process automation.

- We report on the informality of business process modeling languages in Chapter 4 and why this informality is sometimes needed in projects. For example, we report on the semantic inconsistencies of how events are used in today's business process modeling languages in Sect. 4.4.
- We discuss the need for a means to specify arbitrary synchronization in business process models and workflow definitions in Sects. 4.6 and 9.2.10.
- We coin and define the term of a methodology stakeholder in Sect. 5.1. We explain the impact of methodology stakeholders on the software engineering practices of real-world projects.
- Throughout the text we foster a visualization independent viewpoint of business process specification and even more, i.e., a syntax independent viewpoint or to say it better a concrete syntax independent viewpoint – see, e.g., the discussion of abstract syntax in Sect. 5.1.3.
- We describe two different semantics of business processes with multiple start and end events in Sect. 5.2.2, i.e., a self-contained semantics and a global, context-embedded semantics. We describe that the selection of a self-contained semantics has an impact on the flexibility in building hierarchies and try to find an explanation why a self-contained semantics seems often to be preferred in practice.
- We identify the reasons why methodology stakeholders stuck to the guideline of single entry or exit points for business process specifications – see Sect. 5.2.3.
- A visualization-independent characterization of uniqueness of interface points – see Sect. 5.2.4.
- We observe that certain type specifications for data in leveled data flow diagrams are control flow constraints in Sect. 5.3.
- We investigate the opportunity of bringing the best practices of structured programming to the field of business process specification in Chapter 6. This attempt is done in a sophisticated manner. It is accompanied by a reconsideration of the arguments of structured programming in that we ideally target to identify the scientifically discussable core – in the sense of falsifiability [287, 288] – of the structure programming metaphor.
- We explain workflow systems from the viewpoint of human-computer interaction in Sect. 7. We explain workflow systems as three-staged human-computer interaction. On this basis we are able to distinguish between terminal/server-style and windows-style workflow systems and analyze their differences.
- We explain the importance of a general instead of pattern-oriented viewpoint on the assignment of resources to activities in workflow automation in Sect. 7.2.3.
- We identify four well-distinguishable visions for service-oriented architecture, i.e., the enterprise application integration vision, the business-to-business-vision, the flexible processes vision and eventually the software

productizing vision – see Sect. 8.1. This clean distinction can help in projects to identify and prioritize more quickly the actual targets of the different stakeholders who are advocating a service-oriented architecture strategy.
- We identify two different styles of service-oriented architecture for enterprise application architecture which are basically distinguished from each other by whether the service tier implements business logic and holds persistent data and coin the terms fat hub resp. thin hub hub-and-spoke architecture for these architectural styles – see Sect. 8.2.
- We give a characterization of SOA governance as an approach to massive software reuse – see Sect. 8.5.2
- We elaborate that software reuse can be distinguished from software use, i.e., that software reuse is the either a static use of arbitrary software or a dynamic use of multi-tenant software – see Sect. 8.5.3.
- We introduce the notion of a typed business process modeling. This approach has typed workflow charts as a basis which are integrated with a hierarchy of typed business process models – see Sects. 9.2 and 9.4. The analysis of leveled data flow diagrams in Chapter 5 lays the basis for the design of a concrete integrated typed business process platform.
- We introduce workflow charts and define their semantics in Sect. 9.2.2. Workflow charts are typed tripartite directed graphs. Workflow charts extend and generalize formcharts with respect to the needs of executable business process specification. This means that workflow charts resolve the research question posed in Sect. 3.2. Using workflow charts as a domain-specific programming language means closing today's gap in workflow definition and application programming.

Acknowledgements

I am grateful to Prof. Dr. Colin Atkinson, Prof. Dr. Ulrich Brüning, and Prof. Dr. Wolfgang Effelsberg for making this work possible. Thanks to my advisor Colin Atkinson for his deeply impacting scientific remarks. Thanks to Ulrich Brüning for introducing me to the University of Mannheim. Thanks to Ulrich Brüning and Wolfgang Effelsberg for ongoing technical and personal advice.

Thanks to the University of Mannheim for hosting me.

Thanks to Prof. Dr. Josef Küng and Prof. Dr. Gerhard Weiß for reviewing this text. Their comments essentially improved the text.

Innsbruck, July 2010 *Dirk Draheim*

Contents

List of Figures

Listings