PROCESS-AWARE INFORMATION SYSTEMS

Bridging People and Software Through Process Technology

Edited by

MARLON DUMAS
Queensland University of Technology

WIL van der AALST
Eindhoven University of Technology

ARTHUR H. M. ter HOFSTEDE
Queensland University of Technology

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

PROCESS-AWARE INFORMATION SYSTEMS

PROCESS-AWARE INFORMATION SYSTEMS

Bridging People and Software Through Process Technology

Edited by

MARLON DUMAS
Queensland University of Technology

WIL van der AALST
Eindhoven University of Technology

ARTHUR H. M. ter HOFSTEDE
Queensland University of Technology

WILEY-INTERSCIENCE

A JOHN WILEY & SONS, INC., PUBLICATION

Copyright © 2005 by John Wiley & Sons, Inc. All rights reserved.

Published by John Wiley & Sons, Inc., Hoboken, New Jersey.
Published simultaneously in Canada.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, electronic, mechanical, photocopying, recording, scanning, or otherwise, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without either the prior written permission of the Publisher, or authorization through payment of the appropriate per-copy fee to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, (978) 750-8400, fax (978) 750-4470, or on the web at www.copyright.com. Requests to the Publisher for permission should be addressed to the Permissions Department, John Wiley & Sons, Inc., 111 River Street, Hoboken, NJ 07030, (201) 748-6011, fax (201) 748-6008, or online at http://www.wiley.com/go/permission.

Limit of Liability/Disclaimer of Warranty: While the publisher and author have used their best efforts in preparing this book, they make no representations or warranties with respect to the accuracy or completeness of the contents of this book and specifically disclaim any implied warranties of merchantability or fitness for a particular purpose. No warranty may be created or extended by sales representatives or written sales materials. The advice and strategies contained herein may not be suitable for your situation. You should consult with a professional where appropriate. Neither the publisher nor author shall be liable for any loss of profit or any other commercial damages, including but not limited to special, incidental, consequential, or other damages.

For general information on our other products and services or for technical support, please contact our Customer Care Department within the United States at (800) 762-2974, outside the United States at (317) 572-3993 or fax (317) 572-4002.

Wiley also publishes its books in a variety of electronic formats. Some content that appears in print may not be available in electronic format. For information about Wiley products, visit our web site at www.wiley.com.

Library of Congress Cataloging-in-Publication Data:

Process-aware information systems : bridging people and software through
process technology / Marlon Dumas, Wil van der Aalst, Arthur ter Hofstede
(editors).
p. cm.
Includes bibliographical references.
ISBN-13 978-0-471-66306-5
ISBN-10 0-471-66306-9 (cloth : alk. paper)
1. Computer-aided software engineering. 2. Human-computer interaction. I.
Dumas, Marlon. II. Aalst, Wil van der. III. Ter Hofstede, Arthur, 1966–
QA76.758.P757 2005
005.1′0285—dc22

2005001369

Printed in the United States of America.

10 9 8 7 6 5 4 3 2 1

To Inga and her admirable ability to marry reason with emotion—Marlon

To Willem for showing that you do not have to be smart to enjoy life—Wil

Contents

Preface

Process-aware information systems are at the heart of an ongoing "silent revolution." From the late 1970s to the early 1990s, the lion's share of attention in the area of information systems went to data. The focus was mainly on storing and retrieving information and, hence, data models were often the starting point for designing information systems, whereas database management systems were considered to be the heart of the run time infrastructure. During the 1990s, a number of parallel trends shifted the focus to processes. As a result, an increasing number of business processes are now conducted under the supervision of information systems driven by explicit process models. This shift of focus has resulted in a myriad of approaches to process engineering, modeling, and implementation, ranging from those supported by groupware and project management products to those supported by document, imaging, and workflow management systems, which are now finding their way into enterprise application-integration tools. The plethora of (sometimes subtly different) technologies in this area illustrates the relevance of the topic but also its complexity, and despite a number of discontinued and ongoing standardization efforts, there is still a lack of an overarching framework for designing and implementing process-aware information systems. Instead, process-awareness in information systems manifests itself in various forms, with similar concepts appearing under different names, in different combinations, and with varying levels of tool support.

The goal of this book is to provide a unifying and comprehensive overview of the technological underpinnings of the emerging field of process-aware information systems engineering. While primarily intended as a textbook, the book is also a manifesto for process-aware information systems, insofar as it puts forward the resemblances (and differences) between a number of technologies that up to now have evolved somewhat independently of one another. In this respect, it is hoped that the book will raise awareness of the need to look at new trends in the area in light of a broader perspective than has been employed up to now and to draw on the large body of existing theoretical and practical knowledge. In terms of scope, it should be mentioned that the focus of the book is on technical aspects, as opposed to strategic and managerial aspects, which are covered in a number of other publications (many of which are referenced throughout the book).

The book is intended to be used both as a textbook for advanced undergraduate and postgraduate courses and as reference material for practitioners and academics. Consistent with the former purpose, the book contains exercises, ranging from simple questions to projects and possible assignment subjects. Sample solutions for many of these exercises will be made available at a companion site, http://www.wiley.com/WileyCDA/WileyTitle/productCd-0471663069.html. Further information and material related to the book will be posted at: http://www.bpmcenter.org.

The book gathers contributions from a number of international experts and teams from both academia and industry. We acknowledge the contributors for their engagement and dedication in the preparation of their chapters and for their prompt help in peer-reviewing each others' chapters. It should be recognized that many of the topics covered in the book are still emerging or even groundbreaking, and authors had to put considerable effort into presenting them in a way that is accessible to the broadest possible audience. We also acknowledge the financial support of the Australian Research Council through its Discovery Projects scheme. Finally, we thank Wiley's editorial team, especially Val Moliere, for their support and patience that contributed to turning the original book project into a reality.

MARLON DUMAS
WIL VAN DER AALST
ARTHUR H. M. TER HOFSTEDE

Brisbane, Australia,
August 2005

Contributors

Otmar Adam, Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

Gustavo Alonso, Department of Computer Science, ETH Zentrum, Zürich, Switzerland

Paulo Barthelmess, Department of Computer Science, University of Colorado, Boulder, Colorado

Paul J. S. Berens, Pallas Athena, Apeldoorn, The Netherlands

Charles Brown, Logica CMG, Milton, Australia

Christoph Bussler, Digital Enterprise Research Institute, National University of Ireland, Galway, Ireland

Jun Chen, Department of Computer Science, University of Colorado, Boulder, Colorado

Francisco Curbera, Component Systems Group, IBM T.J. Watson Research Center, Hawthorne, New York

Jörg Desel, Catholic University, Faculty of Mathematics and Geography, Eichstätt, Germany

Marlon Dumas, Centre for Information Technology Innovation, Queensland University of Technology, Brisbane, Australia

Clarence A. Ellis, Department of Computer Science, University of Colorado Boulder, Colorado

Gregor Engels, University of Paderborn, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn, Germany

Alexander Förster, University of Paderborn, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn, Germany

Reiko Heckel, University of Paderborn, Faculty of Computer Science, Electrical Engineering and Mathematics, Paderborn, Germany

Rania Khalaf, Component Systems Group, IBM T.J. Watson Research Center, Hawthorne, New York

Jan Mendling, Vienna University of Economics, BA Department of Information Systems New Media Lab, Wien, Austria

Greg Meredith, Microsoft, Seattle, Washington

Nirmal Mukhi, Component Systems Group, IBM T.J. Watson Research Center, Hawthorne, New York

Andreas Oberweis, AIFB, University of Karlsruhe, Karlsruhe, Germany

Adrian Price, Versata, Inc., Oakland, California

Hajo A. Reijers, Eindhoven University of Technology, Department of Technology Management, Eindhoven, The Netherlands

Michael Rosemann, Centre for Information Technology Innovation, Brisbane, Australia

August-Wilhelm Scheer, Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

Arthur H. M. ter Hofstede, Centre for Information Technology Innovation, Queensland University of Technology, Brisbane, Australia

Oliver Thomas, Institute for Information Systems (IWi), German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany

Sebastian Thöne, University of Paderborn, Department of Computer Science, Paderborn, Germany

Wil van der Aalst, Department of Technology Management, Eindhoven University of Technology, Eindhoven, The Netherlands

Alexander Verbraeck, Delft University of Technology, Faculty of Technology, Policy, and Management, Systems Engineering Group, Delft, The Netherlands

Jacques Wainer, Instituto de Computação, Universidade Estadual de Campinas, Caixa, Campinas, Sao Paulo, Brazil

Sanjiva Weerawarana, Component Systems Group, IBM T.J. Watson Research Center, Hawthorne, New York

A. J. M. M. Weijters, Department of Technology Management, Eindhoven University of Technology, Eindhoven, The Netherlands

Michael zur Muehlen, Stevens Institute of Technology, Wesley J. Howe School of Technology Management, Castle Point on Hudson, Hoboken, New Jersey

PART I

CONCEPTS

CHAPTER 1

Introduction

MARLON DUMAS, WIL van der AALST,
and ARTHUR H. M. ter HOFSTEDE

1.1 FROM PROGRAMS AND DATA TO PROCESSES

A major challenge faced by organizations in today's environment is to transform ideas and concepts into products and services at an ever-increasing pace. At the same time and following the development and adoption of Internet technologies, organizations distributed by space, time, and capabilities are increasingly pushed to exploit synergies by integrating their processes in the setting of virtual organizations. These forces triggered a number of trends that have progressively changed the landscape and nature of enabling technologies for information systems development.

Figure 1.1 illustrates some of the ongoing trends in information systems [2]. This figure shows that information systems consist of a number of layers. The center is formed by the system infrastructure, consisting of hardware and the operating system(s) that make the hardware work. The second layer consists of generic applications that can be used in a wide range of enterprises. These applications are typically used in multiple departments within the same organization. Examples of such generic applications are a database management system (DBMS), a text editor, and a spreadsheet editing tool. The third layer consists of domain-specific applications. These applications are only used within specific types of organizations or departments. Examples are decision support systems for vehicle routing, computer-aided design tools, accounting packages, and call center software. The fourth layer consists of tailor-made applications developed for specific organizations.

In the 1960s, the second and third layers were practically missing. Information systems were built on top of a small operating system with limited functionality. Since no generic or domain-specific software was available, these systems mainly consisted of tailor-made applications. Since then, the second and third layers have developed and the ongoing trend is that the four circles are increasing in size, that is, they are moving to the outside while absorbing new functionality. Today's operating systems offer much more functionality, especially in the area of networking.

Process-Aware Information Systems. Edited by Dumas, van der Aalst, and ter Hofstede

Copyright © 2005 John Wiley & Sons, Inc.

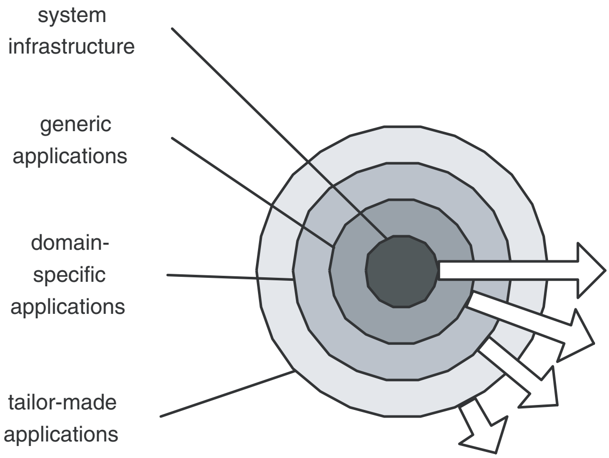

Figure 1.1 Trends relevant to business process management.

DBMSs that reside in the second layer offer functionality that used to be encoded in domain-specific and tailor-made applications. Also, the number and complexity of domain-specific and tailor-made applications has increased, driven by the need to support more types of tasks and users. In addition, the advent of the Web has resulted in these applications being made accessible directly to customers and business partners. The resulting proliferation of applications supporting various tasks and users has engendered a need for a global view on the operation of information systems. Accordingly, the emphasis has shifted from application programming to application integration. The challenge is no longer the coding of individual modules but rather the seamless interconnection and orchestration of pieces of software from all four layers.

In parallel with the trend "from programming to assembling," another trend changed the way information systems were developed. This trend is the shift "from data orientation to process orientation." The 1970s and 1980s were dominated by data-driven approaches. The focus of information technology (IT) was on storing, retrieving, and presenting information primarily seen as data. Accordingly, data modeling was the starting point for building an information system. This led to scalable and robust techniques and tools for developing data-centric information systems. The modeling of business processes, however, was often neglected. As a result, the logic of business processes was spread across multiple software applications and manual procedures, thereby hindering their optimization and their adaptation to changes. In addition, processes were sometimes structured to fit the constraints of the underlying information system, thus introducing inefficiencies such as manual resource allocation and work routing, poor separation of responsibilities, inability to detect work overflows and trigger escalation procedures, unnecessarily batched operations, and redundant data entry steps. Management trends in the early 1990s such as business process reengineering (see Section 1.3.1) brought about an increased emphasis on processes. As a result, system engineers are resorting to more process-driven approaches.

The last trend we would like to mention is the shift from carefully planned designs to redesign and organic growth. Due to the widespread adoption of Internet standards and the connectivity that this engendered, information systems are now required to change within tight deadlines in response to changes in the organization's environment; for example changes in the business focus or the business partners. As a result, fewer systems are built from scratch. Instead, existing applications are partly reused in the new system. Consequently, there is a continuous trend toward software componentization and dynamic and reuse-oriented software engineering approaches—approaches aimed at rapidly and reliably adapting existing software in response to changes in requirements. One of the most recent of these approaches, model-driven architecture (MDA), exploits automated code generation, code refactoring, model transformation, and model execution techniques to achieve a faster turnaround for propagating changes in the design into changes in the implementation.

The confluence of these trends, which are summarized in Figure 1.1, has set the scene for the emergence of an increasing number of *process-aware information systems* (PAISs). PAISs are built on top of a technological infrastructure that can take the form of separate applications residing in the second layer or integrated components in the third layer. Notable examples of PAIS infrastructure residing in the second layer are workflow management systems, process-aware groupware, and some enterprise application integration (EAI) platforms (see discussion in Section 1.3). The idea of isolating the management of processes in a separate component is consistent with the three trends discussed above. PAIS infrastructures can be used to avoid hard-coding the processes into tailor-made applications and thus support the shift from programming to assembling. Moreover, process awareness in both manual and automated tasks is supported in a way that allows organizations to efficiently manage their resources. Finally, pulling away the process logic from application programs and capturing this logic in high-level models facilitates redesign and organic growth. For example, today's workflow management systems and EAI platforms enable designers and developers to implement process change by working on diagrammatic representations of process models, a practice consistent with MDA. In addition, isolating the management of processes in a separate component is consistent with recent developments in the domain of intra- and interorganizational application integration (e.g., emergence of Web services and service-oriented architectures).

1.2 PAIS: DEFINITION AND RATIONALE

As illustrated by Figure 1.1, there has been a shift from data orientation to process orientation, triggering the development of PAISs. Since PAISs can be seen as special kinds of information system, we first discuss the term *information system.* Alter [6] provides the following definition of the term information system: "An *informa-*

tion system is a particular type of *work system* that uses *information technology* to capture, transmit, store, retrieve, manipulate, or display information, thereby supporting one or more other work systems." This definition uses two key terms: *information technology* and *work system.* Alter defines information technology as "the hardware and software used to [store, retrieve, and transfer] information," and a work system as "a system in which human participants perform a business process using information, technology, and other resources to produce products for internal customers."

Figure 1.2 depicts Alter's framework for information systems [6]. It shows an integrated view of an information system encompassing six types of entities: customers, products, business process, participants, information, and technology. The customers are the actors that interact with the information system through the exchange of products (or services). These products are being manufactured/assembled in a business process that uses participants, information, and technology. Participants are the people that do the work. Information may range from information on customers to information about the process. Technology is used in the business process to enable new ways of doing work. Diagrams like the one shown in Figure 1.2 always trigger a discussion on the scope of an information system. Some will argue that all six elements constitute an information system, whereas others will argue that only a selected subset (e.g., just business process, information, and technology) constitute an information system. In this chapter, we do not decide on a single definition of "information system" but use the term in different (although related) senses depending on the context. This book considers a specific type of information systems, that is, information systems that are process aware, and therefore link information technology to business processes. By process, we mean a way for an organizational entity to "organize work and resources (people, equipment, in-

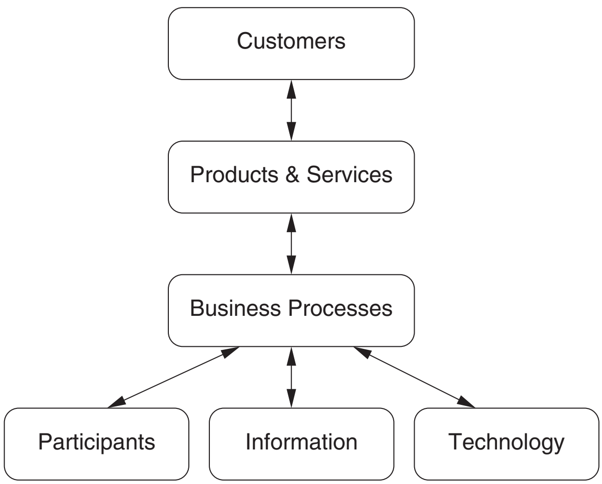

Figure 1.2 An integrated view of an information system.

formation, and so forth) to accomplish its aims" [23]. Sometimes, processes within an organization are hidden—they only manifest themselves in the way people and application programs interact with each other, without being driven by an a priori conception of the way work should be conducted. Other times, processes are captured as a priori defined (i.e., explicit) process models that are used to guide them or even to automate them.

Given these considerations, this book adopts the following definition of a PAIS: *a software system that manages and executes operational processes involving people, applications, and/or information sources on the basis of process models.* Although not part of the adopted definition, it can be noted that these process models are usually represented in a visual language, for example, a Petri net-like notation (Chapter 7). The models are typically instantiated multiple times (e.g., for every customer order) and every instance is handled in a predefined way (possibly with variations).

Given this definition, one can see that a text editor is not "process aware" insofar as it is used to facilitate the execution of specific tasks without any knowledge of the process of which these tasks are part. A similar comment can be made of an e-mail client. A task in a process may result in an e-mail being sent, but the e-mail client is unaware of the process it is used in. At any point in time, one can send an e-mail to any person without being supported or restricted by the e-mail client. Text editors and e-mail clients (at least contemporary ones) are applications supporting tasks, not processes. The same applies to a large number of applications used in the context of information systems.

The shift from task-driven to process-aware information systems brings a number of advantages:

- The use of explicit process models provides a means for communication between managers and business analysts who determine the structure of the business process, and the IT architects, software developers, and system administrators who design, implement, and operate the technical infrastructure supporting these processes.
- The fact that PAISs are driven by models rather than code allows for changing business processes without recoding parts of the systems, that is, if an information system is driven by process models, only the models need to be changed to support evolving or emerging business processes [3].
- The explicit representation of the processes supported by an organization allows their automated enactment [1, 17, 20]. This, in turn, can lead to increased efficiencies by automatically routing information to the appropriate applications and human actors, prioritizing tasks according to given policies, optimizing the time and resources required to deliver services to users, and so on. Also, providing a global view on the operations supported by an information system enables the reduction of redundant data entry tasks and provides opportunities for interconnecting otherwise separate transactions.
- The explicit representation of processes enables management support at the (re)design level, that is, explicit process models support (re)design efforts

[22]. For example, verification tools such as Woflan[1] allow for the verification of workflow models exported from tools such as Staffware[2] (see Chapter 14), ARIS,[3] and Protos.[4] Other tools allow for the simulation of process models. Simulation is a useful tool for predicting the performance of new processes and evaluating improvements to existing processes.

- The explicit representation of processes also enables management support at the control level. Generic process monitoring facilities provide useful information about the process as it unfolds. This information can be used to improve the control of the process, for example, moving resources to the bottleneck in the process. Recently, process monitoring has become one of the focal points of BPM vendors, as reflected by product offerings such as ARIS Process Performance Monitor (PPM) of IDS Scheer[5] and OpenView Business Process Insight (BPI) of HP.[6] This trend has also triggered research into workflow mining (Chapter 10) and process execution analysis and control [8, 25].

1.3 TECHNIQUES AND TOOLS

1.3.1 A Historic View of PAISs

To better understand the emergence and adoption of PAISs and their associated techniques and tools, it is insightful to take a quick historic overview. An interesting starting point, at least from a scientific perspective, is the early work on process modeling in office information systems by Skip Ellis [10], Anatol Holt [16], and Michael Zisman [24]. These three pioneers of the field independently applied variants of Petri net formalism (see Chapter 7) to model office procedures. During the 1970s and 1980s, there was great optimism in the IT community about the applicability of office information systems. Unfortunately, few applications succeeded, in great part due to the lack of maturity of the technology, as discussed below, but also due to the existing structure of organizations, which was primarily centered around individual tasks rather than global processes. As a result of these early negative experiences, both the application of this technology and related research almost stopped for nearly a decade. Hardly any advances were made after the mid-1980s. Toward the mid-1990s, however, there was a renewed interest in these systems. Instrumental in this revival of PAISs was the popularity gained (at least in managerial spheres) by the concept of *business process reengineering* (BPR) advocated by Michael Hammer [14, 15] and Thomas Davenport [9], among others. The idea promoted by BPR is that overspecialized tasks carried across different organizational

[1]http://www.tmis.tue.nl/research/woflan
[2]http://www.tibco.com/company/staffware.jsp
[3]http://www.ids-scheer.com
[4]http://www.pallas-athena.com
[5]http://www.ids-scheer.com
[6]http://www.hp.com

units need to be (re)unified into coherent and globally visible processes. In particular, IT should not only support the automation of individual tasks, but should also be seen as an instrument for coordinating and interconnecting tasks and resources (e.g., people, physical assets, software applications).

In the aftermath of the BPR wave, and despite some (sometimes well-founded) criticisms and early failures in the implementation of the underlying concepts, the importance of PAISs grew steadily. The early and mid-1990s saw the advent of business process modeling tools such as Protos and ARIS, as well as workflow management systems such as FlowMark [19][7] and Staffware. The number of PAIS-related tools that have been developed in the past decade and the continuously increasing body of professional and academic literature in this field of technology is overwhelming. Today's off-the-shelf workflow management systems and business process modeling tools are readily available. However, their application is still limited to specific industries such as banking and insurance. As pointed out by Skip Ellis [11], it is important to learn from the ups and downs of PAIS-related technologies. The failures in the 1980s can be explained by both technical and conceptual factors. In the 1980s, networks were slow, expensive, or not present at all; the development of suitable graphical interfaces was hindered by hardware limitations; and application developers were concentrated on addressing other problems such as scalable data storage and retrieval. At the same time, there were also more conceptual problems such as: (i) a lack of a unified way of modeling processes, (ii) a lack of methods for seamlessly propagating changes in the requirements into changes in the design and then into changes in the implementation, and (iii) the systems were too rigid to be used by people in the workplace. Most of the technical limitations have been more or less satisfactorily resolved by now. However, the more conceptual problems remain. In particular, widely adopted and unambiguous standards for business process modeling are still missing, and even today's workflow management systems enforce unnecessary constraints on the process logic (e.g., processes are made more sequential than they need to be). This book will discuss some of the traditional process models (e.g., Petri nets) and some of the emerging standards (e.g., BPEL). However, there is no consensus on which models and standards to use. New paradigms such as case handling (see Chapter 15) and associated products such as FLOWer offer more flexibility but still only provide a partial solution to the many problems related to the alignment of people, processes, and systems.

1.3.2 PAIS Development Tools

There are basically two ways to develop a PAIS: (i) develop a specific process support system, or (ii) configure a generic system. In the first case, an organization builds its own process support system "from scratch" with the specific aim of supporting its processes. This organization-specific system can be as simple as a soft-

[7]FlowMark was later integrated into the message-oriented middleware platform MQSeries to become MQSeries Workflow. Subsequently, this platform was renamed WebSphere MQ, so that the workflow system is currently known as "Websphere MQ Workflow."

ware library providing routines for incorporating process awareness into applications, or it can take the form of a process execution platform providing facilities for defining, testing, deploying, executing, and monitoring a large class of processes. This ad hoc approach ensures that the resulting system fits the needs of the organization and the specificities of its processes. However, the initial investment cost of this approach may be too high for some organizations, and the resulting system may not be scalable. As new processes are introduced, existing processes become more sophisticated, and users develop higher expectations, it becomes difficult to adapt the process support system to meet new demands.

Generic process support systems, on the other hand, are generally not developed by organizations actually using a PAIS (although there are cases in which an organization-specific system has subsequently evolved into a system comparable to a generic software product). A typical example of a generic software product is a workflow management system (WFMS) such as Staffware. WFMSs are generic in that they do not incorporate information about the structure and processes of any particular organization. Instead, to use such a generic system, an organization needs to configure it by specifying processes, applications, organizational entities, and so on. These specifications are then executed by the generic system. In the case of a WFMS, when certain types of events occur (e.g., arrival of a purchase order), an instance of the relevant process (called a *workflow*) is triggered, and this results in one or several tasks being enabled. Enabled tasks are then routed to people or applications who/which complete them. As tasks are completed, the WFMS proceeds by dispatching more tasks as per the process specification, until the process instance is completed.

At present, there are more than one hundred WFMSs. A typical workflow management system is composed of a design tool, an execution engine, a worklist management system, adapters for invoking various types of applications, and, in a few cases, modules for monitoring, auditing, and analyzing existing workflow models.

Although the classical apparatus for developing PAISs is workflow technology, “pure WFMSs” are far from being the only type of tool used for developing PAISs. Process awareness is also supported in different ways by the following types of tools:

- Process-aware collaboration tools such as Caramba (see Chapter 2).
- Project management tools such as AMS Realtime[8] and Microsoft[9] Project.
- Tracking tools (e.g., for job, issue, or call tracking) such as JobPro Central.[10]
- Enterprise resource planning (ERP) and customer relationship management (CRM) systems such as SAP[11] and Peoplesoft,[12] which incorporate a workflow management system within a broader enterprise system management solution.

[8]http://www.amsrealtime.com
[9]http://www.microsoft.com
[10]http://www.jobprocentral.com
[11]http://www.sap.com
[12]http://www.peoplesoft.com

- Case handling systems such as FLOWer (see Chapter 15).
- Business process design and engineering tools such as ARIS and Protos.
- Enterprise Application Integration (EAI) suites such as TIBCO[13] ActiveEnterprise and Microsoft BizTalk.
- Extended Web application servers (also called Web integration servers) such as BEA[14] WebLogic Integration and IBM[15] Websphere MQ.

Furthermore, process support may be found in various forms outside the realm of information systems. For instance, the emergence of process-centered software engineering environments (PSEEs) [13] illustrates that process awareness can be beneficial in other domains where people and applications need to interact in a coordinated manner.

The plethora of similar but subtly different enabling technologies for process-aware information systems is overwhelming. On the one hand, this demonstrates the practical relevance of process support. On the other hand, it illustrates that process support is far from trivial. At present, there is a "Babel of approaches" to deal with process awareness in information systems. This is hindering the emergence and general understanding of the common principles underlying these approaches.

1.4 CLASSIFICATIONS

A starting point from which to build a structured view on the landscape of supporting techniques, technologies, and tools for PAISs is to classify them according to orthogonal dimensions. The following subsections introduce and illustrate some of these dimensions.

1.4.1 Design-Oriented Versus Implementation-Oriented

Figure 1.3 summarizes the phases of a typical PAIS life cycle. In the design phase, processes are designed (or redesigned) based on a requirements analysis, leading to process models. In the implementation (or configuration) phase, process models are refined into operational processes supported by a software system. This is typically achieved by configuring a generic infrastructure for process-aware information systems (e.g., a WFMS, a tracking system, a case handing system, or an EAI platform). After the process implementation phase (which encompasses testing and deployment), the enactment phase starts—the operational processes are executed using the configured system. In the diagnosis phase, the operational processes are analyzed to identify problems and to find aspects that can be improved.

Different phases of the PAIS life cycle call for different techniques and types of tools. For example, the focus of traditional WFMSs is on the lower half of the PAIS

[13] http://www.tibco.com
[14] http://www.bea.com
[15] http://www.ibm.com

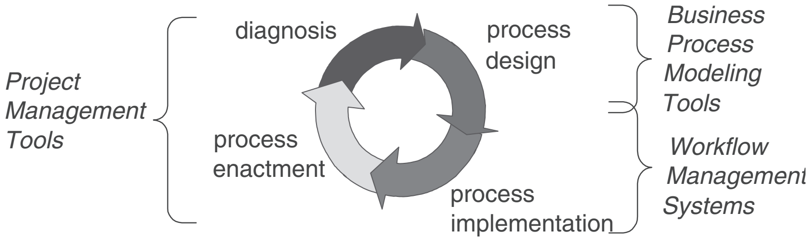

Figure 1.3 The PAIS life cycle.

life cyle. They are mainly aimed at supporting process implementation and execution and provide little support for the design and diagnosis phases. Indeed, although WFMSs are able to log process-related data, they rarely provide tools for real-time and offline interpretation of these data. There are some research proposals in the area of process-related data analysis (e.g., the Process Data Warehouse [7] and the Business Process Cockpit [8]) but these have made their way into commercial products only in a limited way (e.g., ARIS PPM and HP Openview BPI mentioned above). Moreover, support for the design phase is limited to providing a graphical editor, whereas model analysis (e.g., through simulation and static verification) and methodological support are missing.

At the other end of the spectrum, business process modeling tools are design-oriented, focusing on the top half of the PAIS lifecycle. For instance, ARIS (Chapter 6) supports a reuse-oriented design methodology by providing libraries of reference models that may be adapted to meet the needs of specific organizations.

Other types of PAIS-related tools (e.g., project management tools) are hybrid in the sense that they support both design (e.g., PERT and resource allocation analysis) and execution (e.g., Web-based project tracking). However, these hybrid tools tend to focus on very specific types of processes (e.g., projects, job handling in IT help desks, customer call handling). In a way, these tools may be seen as "vertical PAIS development tools," in that they cover a large section of the PAIS development life cycle, but do so by restricting their scope to specific problem domains.

1.4.2 People Versus Software Applications

Another way of classifying PAISs is in terms of the nature of the participants (or resources) they involve and, in particular, whether these participants are humans or software applications. In this respect, PAISs can be classified into human-oriented and system-oriented [12] or, more precisely, into person-to-person (P2P), person-to-application (P2A), and application-to-application (A2A) processes.

In P2P processes, the participants involved are primarily people, that is, the processes primarily involve tasks that require human intervention. Job tracking, project management, and groupware tools are designed to support P2P processes. Indeed, the processes supported by these tools usually do not involve entirely automated tasks carried out by applications. Also, the applications that participate in