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.

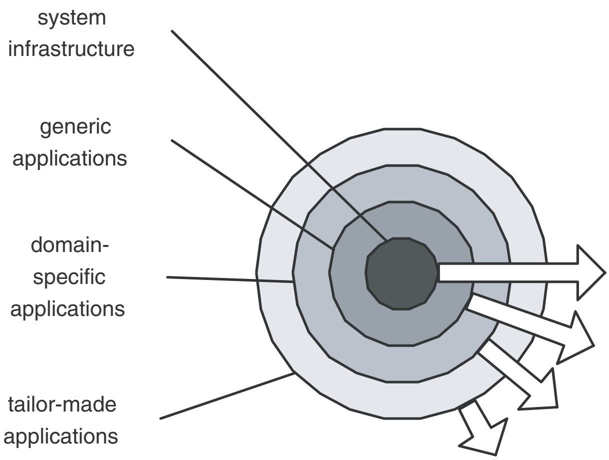

Figure 1.1 Trends relevant to business process management.

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-

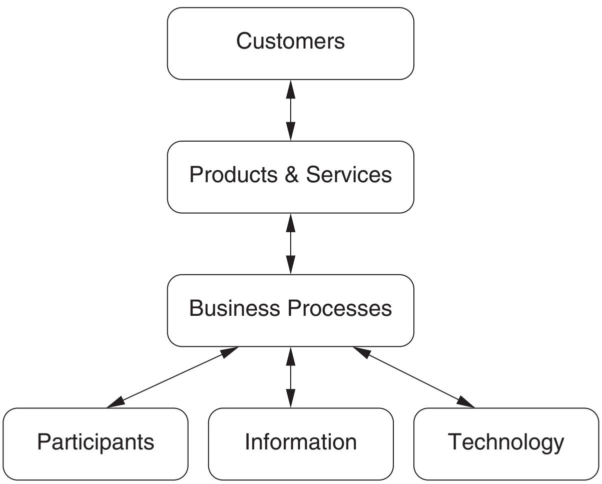

Figure 1.2 An integrated view of an information system.

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

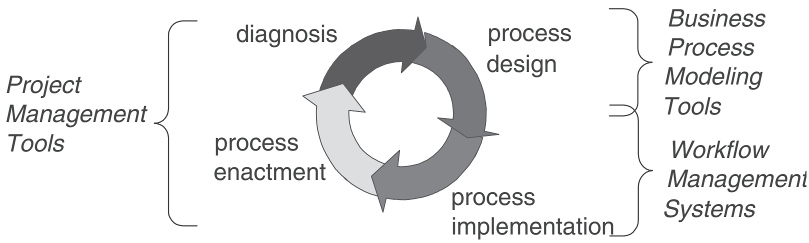

Figure 1.3 The PAIS life cycle.

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 these processes (e.g., project tracking servers, e-mail clients, video-conferencing tools, etc.) are primarily oriented toward supporting computer-mediated interactions.

At the other end of the spectrum, A2A processes are those that only involve tasks performed by software systems. Such processes are typical in the area of distributed computing and, in particular, distributed application integration. Transaction processing systems, EAI platforms, and Web-based integration servers are designed to support A2A processes. It should be noted that sometimes the logic of these processes is captured by explicit process models, and other times it is implicitly coded into the programs that participate in the process. As the resources participating in A2A processes are applications, and these may share common databases, an important aspect that arises in this type of process is ensuring certain transactional properties as defined in the realm of database management systems (DBMSs). Techniques relevant to this aspect are presented in Chapter 11.

Finally, P2A processes are those that involve both human tasks and interactions between people, and tasks and interactions involving applications that act without human intervention. Workflow systems fall in the P2A category since they primarily aim at making people and applications work in an integrated manner. Note that since workflow systems support both people and applications, they can also be used to support interactions between people only, as well as interactions between applications only. A workflow system can, in principle, be used as a platform to implement A2A processes, although it may be preferable in these situations to use a platform specifically designed for this purpose. On the other hand, pure manufacturing workflow may be considered to be P2P rather than P2A. However, most workflow products nowadays support interactions between both people and applications and, therefore, we consider workflow technology as a whole to be P2A.

The boundaries between P2P, P2A, and A2A are not crisp. Instead, there is a continuum of techniques and tools from P2P (i.e., manual, human-driven) to A2A (automated, application-driven). In particular, ad hoc process and case-handling systems (see Chapters 2 and 15) can be placed in between the P2P and P2A categories. On the other hand, some tools target both A2A and P2A systems. For example, the IBM Websphere MQ family supports both application integration and workflow management.

1.4.3 Structure and Predictability of Processes

The degree of structure of the process to be automated (which is strongly linked to its predictability) is frequently used as a dimension to classify PAISs. In this respect, a traditional distinction is that between *ad hoc, administrative,* and *production processes* [21, 12]. An ad hoc process is one in which there is no a priori identifiable pattern for moving information and routing tasks among people; for example, a product documentation process or a process for preparing a response to a complex tender. Administrative processes, on the other hand, involve predictable processes with relatively simple task coordination rules. These rules may be revised with some frequency or may be adapted to fit exceptional cases, but, in any case, they capture the core of the process. Finally, production processes involve repetitive and predictable tasks with more or less complex but highly stable task coordination rules.

The above classification mixes the predictability of the process with its complexity. As process modeling has matured, it has become evident that some administrative processes can be relatively complex. A slightly different classification that considers only the predictability aspect is that between *unframed, ad hoc framed, loosely framed,* and *tightly framed* processes [4]. A process is said to be unframed if there is no explicit process model associated with it. This is the case for collaborative processes supported by groupware systems that do not offer the possibility of defining process models. Unframed processes are out of the scope of this book, although they are referenced in some parts (e.g., Chapter 2) insofar as unframed processes can lead to framed ones, and there is no clear-cut boundary between these categories.

A process is said to be ad hoc framed if a process model is defined a priori but only executed once or a small number of times before being discarded or changed. This is the case in project management environments in which a process model (i.e., a project chart) is often only executed once. It is also the case in grid computing environments in which a scientist may define a process model corresponding to a computation involving a number of datasets and computing resources, and then run this process only once (a type of process also known as *scientific workflows* or *grid workflows*). Chapter 2 provides an overview of a system designed to support ad hoc processes (Caramba).

A loosely framed process is one for which there is an a priori defined process model and a set of constraints, such that the predefined model describes the "normal way of doing things" while allowing the actual executions of the process to deviate from this model within certain limits. In other words, the trajectory of a process instance is restricted by some upper and lower bound. Case handling systems such as FLOWer support loosely framed processes by allowing implicitly specified routes. Ad hoc workflow systems such as TIBCO InConcert allow for adaptations of a process template or emerging processes such that every execution can be seen as corresponding to a different process model. In other words, the a priori defined process model is implicitly adapted to suit the requirements of each case.

Finally, a tightly framed process is one that consistently follows an a priori defined process model. This is the case of traditional workflow systems, of which Staffware (Chapter 18) is an example.

As with P2P, P2A, and A2A processes, the boundaries between unframed, ad hoc framed, loosely framed, and tightly framed processes are not crisp. In particular, there is a continuum between loosely and tightly framed processes. For instance, during its operational life a process considered to be tightly framed can start deviating from its model so often and so unpredictably that at some point in time it may be considered to have become loosely framed. Conversely, after a large number of cases of a loosely framed process have been executed, a common structure may become apparent, which may then be used to frame the process in a tighter way. Process mining techniques (see Chapter 12) provide a means for discovering such a "common structure" in a large number of process cases.

Figure 1.4 plots different types of PAISs and PAIS-related tools with respect to the degree of framing of the underlying processes (unframed, ad hoc, loosely, or tightly framed), and the nature of the process participants (P2P, P2A, and A2A).

1.4.4 Intraorganizational Versus Interorganizational

Initially, process-aware information systems were mainly oriented towards intraorganizational settings. Focus was on the use of process support technologies (e.g., workflow systems) to automate operational processes involving people and applications inside an organization (or even within an organizational unit). Over the last few years, there has been a push toward processes that cross organizational barriers. Such interorganizational processes can be one-to-one (i.e., bilateral relations), one-to-many (i.e., an organization interacting with several others), or many-to-many (i.e., a number of partners interacting with each other to achieve a common goal).

The trend toward interorganizational PAISs is marked by the emergence of *business-to-business (B2B) integration standards* that define collections of common B2B integration processes (e.g., for procurement) or support the definition of such processes (see Chapter 4). It is also apparent in the emergence of the notion of *(Web) service composition,* whereby applications are exported as services and composed by means of process models [5]. This notion is embodied in standards such as (WS-)BPEL (see Chapter 13) and WS-CDL [18]. A number of tools implementing these standards (or subsets thereof) are now emerging, and established tools for intraorganizational application and process integration are being extended to support these standards.

The modeling of collaborative interactions as explicit process models is a central issue in B2B integration (see Chapter 4). In this area, processes appear in two

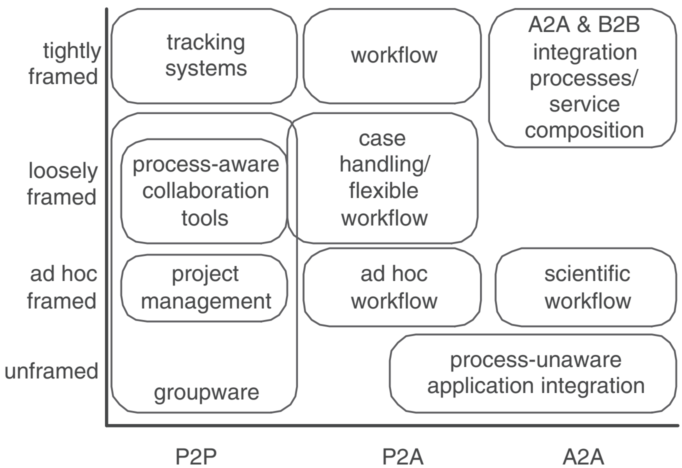

Figure 1.4 Types of PAISs and associated development tools.

forms: public and private. Public processes are those whose definitions are visible to parties outside the organization which implements the process. On the other hand, the definition of a private process is only visible to the organization that owns it. The rationale behind this distinction is twofold. On the one hand, organizations do not wish to expose the full details of their processes to other organizations. Instead, they would only expose the parts of the process that are relevant for establishing a given collaboration. On the other hand, it allows for partners to be replaced. An organization A partnering with an organization B in the context of an interorganizational process is able to substitute B for another organization C, so long as C provides a public process compatible with the requirements of A.

1.5 ABOUT THE BOOK

1.5.1 Goal and Intended Audience

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.* To achieve this goal, the book brings together contributions from leading experts in related fields. These contributions have been selected because they complement each other and cover some of the most salient aspects of the overall picture of process-aware information systems.

Building, deploying, and running a process-aware information system, especially in a mid- or large-scale environment, is a daunting task. It often involves a considerable number of stakeholders. These range from the chief technology officers, chief process officers, and/or managers who set the strategic directions for process automation, (re-)deployment, change, or continuous improvement projects, through the business analysts and IT architects who define the requirements and high-level specifications of the system, down to the process designers and application developers who refine the higher-level specifications into a deployable system. To this list should be added the most important actors: the users who interact with the system in their everyday conduct of business, as well as the operations managers, system administrators, and IT helpdesk assistants who ensure the day-to-day running and ad hoc troubleshooting of the system. The variety of involved stakeholders gives an idea of the multidisciplinarily nature of process-aware information systems engineering. This book does not intend to cover all aspects of this field. Instead, it focuses on technological aspects. Business and social aspects are only addressed when required to illustrate possible uses of certain techniques, technologies, or tools. Furthermore, the book does not directly address methodological issues although it refers to best practices in applying specific techniques.

The book is primarily intended for advanced students specializing in information systems technologies. It is designed to be used as a textbook for a one-semester, topic-oriented course on business process management, business process engineering, or workflow. It may equally well serve as a reference book for a course on enterprise systems. The book is also targeted at professionals involved in projects related to process-aware information systems, including business process modeling, workflow, groupware and teamwork, enterprise application integration, and business-to-business integration. In addition, since the book covers both practical and theoretical approaches to process support, it should also be of great interest to researchers and research students.

To support its pedagogic goal, chapters are structured in the style of tutorials. They present general aspects before zooming into specific technical issues. In addition, the book contains numerous examples and graphical illustrations, and each chapter includes a collection of thought-provoking questions and exercises of varying degrees of difficulty, allowing the reader to review major concepts and techniques. Solutions to most of these exercises are provided on the book's companion website. Finally, the book contains a list of resources including suggested readings as well as pointers to relevant portals, standardization bodies, initiatives, and consortia. These references complement those provided at the end of each chapter.

1.5.2 Overview of Contents

The book is divided into four parts. Part I exposes and illustrates some foundational concepts of PAISs. It also provides an overview of languages, techniques, standards and tools, but without entering into the level of detail of subsequent parts. In addition to the present chapter, this part includes three other chapters corresponding to the classification of PAISs according to the nature of the participants (i.e., P2A, P2P, A2A) as discussed in Section 1.4.2. Chapter 2 opens with an overview of P2A processes as embodied in WFMSs. This discussion of "mainstream" technology lays the ground for the discussions on P2P processes (Chapter 3) and A2A and B2B (Business-to-Business) processes (Chapter 4), which cover more "avant-garde" technology, reflecting the fact that for a long time process-awareness in information systems has been considered mainly in the setting of systems that intertwine human and automated tasks and the focus is now progressively expanding into more human-centric and system-centric processes.

Part II is dedicated to process modeling languages. Chapter 5 shows how UML, a widely adopted object-oriented modeling standard, can be applied to (business) process modeling. The authors demonstrate that the various types of diagrams included in the UML standard provide the building blocks for modeling processes, but that in order to apply them to the domain of process modeling, it is important to understand their overlap and how they complement each other. Chapter 6 presents the extended event-driven process chains (eEPCs) notation. In contrast to UML which is general-purpose, eEPCs are specifically designed to support business process modeling. They are supported by a well-known tool called ARIS, which provides a range of functionality for designing and analyzing business processes. To complement the overviews of UML and EPCs, two modeling languages widely used in practice, Chapter 7 looks at a formal notation for process modeling, namely Petri nets. This formal notation has been applied to a wide variety of domains such as concurrent systems analysis, communication networks design, critical systems verification, and workflow modeling. Several business process modeling and execution languages (or subsets thereof) have been given semantics in terms of Petri nets, including UML activity diagrams (Chapter 5) and BPEL (Chapter 13). There are also several products that directly support Petri nets, for example, workflow systems such as COSA[16] and Promatis[17] INCOME as well as business process modeling tools such as Protos. Part II closes with Chapter 8, which presents a set of patterns that have been used to evaluate the capabilities and limitations of a number of workflow specification languages (their original scope) but also process modeling and service composition languages.

Part III presents techniques relevant to the development of PAISs. As with the rest of the book, the intention is not to be exhaustive in terms of coverage. Instead, an in-depth presentation of techniques in selected areas is provided, namely process design, process mining, and transactional process development. Chapter 9 deals with issues at the frontier between the managerial and the technological views of PAISs. The methods and techniques introduced in this chapter are notably relevant in the design phase of the PAIS development lifecycle (Figure 1.3). When starting from scratch, business requirements can be mapped into process models. For existing process models, their alignment with the requirements could be improved with these techniques, in particular in terms of performance. Chapter 10 presents techniques that are relevant to the diagnosis phase of the PAIS life cycle. Specifically, it presents a set of techniques for automatically unveiling knowledge about the structure of process executions by analyzing event logs gathered during these executions. These techniques make it possible to identify discrepancies between the way processes are expected to execute (as captured in the corresponding process models) and the way they actually execute. Part III closes with Chapter 11, which deals with transaction management, and discusses how this aspect emerges in the context of business process execution.

To close the book, Part IV focuses on the application of the concepts, modeling approaches, and techniques presented in the previous parts by showing how some of them are embodied in specific standards and tools. Chapter 12 provides an overview of standards developed by the Workflow Management Coalition. These standards consolidate a number of concepts, language constructs, and interfaces supported by WFMSs. Chapter 13 presents a more recent standardization effort in the area of A2A processes, namely the Business Process Execution Language for Web Services (WS-BPEL or BPEL for short). Finally, Chapters 14 and 15 present two PAIS development tools. The first one, Staffware, is a representative of tightly framed P2A process development tools, whereas the second one, FLOWer, is intended to support loosely framed P2A processes, with some features relevant for P2P processes (in particular regarding work authorization and distribution). In line with the spirit of the book, these closing chapters do not focus on how to use the presented tools, but rather on how these tools provide realizations of general concepts and principles, as well as how they may be used to address PAIS development challenges.

[16]http://www.cosa.nl
[17]http://www.promatis.de/english

REFERENCES

1. W. M. P. van der Aalst and K. M. van Hee. *Workflow Management: Models, Methods, and Systems.* MIT Press, Cambridge, MA, 2002.
2. W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske. Business Process Management: A Survey. In W. M. P. van der Aalst, A. H. M. ter Hofstede, and M. Weske, editors, *International Conference on Business Process Management (BPM 2003),* volume 2678 of *Lecture Notes in Computer Science,* pp. 1–12. Springer-Verlag, Berlin, 2003.
3. W. M. P. van der Aalst and S. Jablonski. Dealing with Workflow Change: Identification of Issues and Solutions. *International Journal of Computer Systems, Science, and Engineering, 15*(5):267–276, 2000.
4. W. M. P. van der Aalst, M. Stoffele, and J. W. F. Wamelink. Case Handling in Construction. *Automation in Construction, 12*(3):303–320, 2003.
5. G. Alonso, F. Casati, H. Kuno, and V. Machiraju. *Web Services: Concepts, Architectures and Applications.* Springer-Verlag, Berlin, 2003.
6. S. Alter. *Information Systems: A Management Perspective.* Addison-Wesley, Reading, MA, 1999.
7. F. Casati. Intelligent Process Data Warehouse for HPPM 5.0. Technical Report HPL-2002-120, HP Labs, 2002.
8. M. Castellanos, F. Casati, U. Dayal, and M.-C. Shan. A comprehensive and automated approach to intelligent business processes execution analysis. *Distributed and Parallel Databases, 16*(3):239–274, 2004.
9. T. H. Davenport. *Process innovation: Reengineering Work through Information Technology.* Harvard Business School Press, Boston, 1992.
10. C. A. Ellis. Information Control Nets: A Mathematical Model of Office Information Flow. In *Proceedings of the Conference on Simulation, Measurement and Modeling of Computer Systems,* pp. 225–240, Boulder, CO, 1979. ACM Press.
11. C. A. Ellis and G. Nutt. Workflow: The Process Spectrum. In A. Sheth, editor, *Proceedings of the NSF Workshop on Workflow and Process Automation in Information Systems,* pp. 140–145, Athens, GA, May 1996.
12. D. Georgakopoulos, M. Hornick, and A. Sheth. An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. *Distributed and Parallel Databases, 3*(2):119–153, 1995.
13. V. Gruhn. Process-Centered Software Engineering Environments, A Brief History and Future Challenges. *Annals of Software Engineering, 14*(1–4):363–382, 2002.
14. M. Hammer. Reengineering Work: Don't automate, Obliterate. *Harvard Business Review,* pp. 104–112, July/August 1990.
15. M. Hammer and J. Champy. *Reengineering the Corporation: A Manifesto for Business Revolution.* Nicolas Brealey Publishing, London, 1993.
16. A. W. Holt. Coordination Technology and Petri Nets. In G. Rozenberg, editor, *Advances in Petri Nets 1985,* volume 222 of *Lecture Notes in Computer Science,* pp. 278–296. Springer-Verlag, Berlin, 1985.
17. S. Jablonski and C. Bussler. *Workflow Management: Modeling Concepts, Architecture, and Implementation.* International Thomson Computer Press, London, UK, 1996.
18. N. Kavantzas, D. Burdett, and G. Ritzinger. Web Services Choreography Description Language Version 1.0. W3C Working Draft, http://www.w3.org/TR/wscdl-10, 2004.

19. F. Leymann and W. Altenhuber. Managing Business Processes as an Information Resource. *IBM Systems Journal, 34*(2):326–348, 1994.
20. F. Leymann and D. Roller. *Production Workflow: Concepts and Techniques.* Prentice-Hall PTR, Upper Saddle River, NJ, 1999.
21. S. McCready. There is More Than one Kind of Workflow Software. *Computerworld, 2:*86–90, November, 1992.
22. H. A. Reijers. *Design and Control of Workflow Processes: Business Process Management for the Service Industry,* volume 2617 of *Lecture Notes in Computer Science.* Springer-Verlag, Berlin, 2003.
23. A Sharp and P. McDermott. *Workflow Modeling: Tools for Process Improvement and Application Development.* Artech House, Norwood, MA, 2001.
24. M. D. Zisman. *Representation, Specification and Automation of Office Procedures.* Ph.D. thesis, University of Pennsylvania, Wharton School of Business, 1977.
25. M. zur Muehlen. *Workflow-based Process Controlling: Foundation, Design and Application of Workflow-driven Process Information Systems.* Logos Verlag, Berlin, 2004.