Workflow Management
Models, Methods, and Systems

Wil van der Aalst and Kees van Hee

The MIT Press Cambridge, Massachusetts London, England
5

Functions and Architecture of Workflow Systems

5.1 Role of Workflow Management Systems

Extensive attention has been paid in the previous chapters to modeling and improving business processes. Techniques were presented for describing these in a structured way, for analyzing them, and for improving them. Clearly these techniques are the key to achieving drastic improvements in the efficiency and effectiveness of the organization and its work performance. One obvious question is how we can realize the desired business process using information technology. In doing so, we must not lose sight of the benefits of a process-oriented approach. The information system must be structured in such a way that it can respond to possible future changes. In practice, this means that information systems must meet a number of requirements:

- Information systems must be set up in such a way that the structure of the business processes is clearly reflected in them. This makes the process recognizable to the user and reduces the chances of errors occurring both during the development of the system and during the execution of the process.
- There should be an integrated approach, which also encompasses non computerized tasks. Today's business processes now frequently extend far beyond what traditionally has been recorded in an information system.
- Information systems must be set up in such a way that the structure of the business processes can be modified easily. This enables organizations to respond flexibly to their changing environment and to restructure their business processes accordingly.
- It is important that the performance of a business process can be tracked properly so that any problems can be discovered at an early stage. Interventions should also be straightforward and possible at the moment.
when something goes wrong. To this end, the performance of the business process should be easy to measure, and it should be possible to refine that performance.

- The allocation of work to people is a point of particular interest. Good workload management is crucial to achieving effective and efficient business processes.

5.1.1 How information systems are traditionally structured
Traditionally, process management has not been separated from the application software in information systems. In other words, the process management has been hidden inside the information system. Because very little attention has been paid to process structure within the framework of traditional systems, it often has been difficult to actually recognize the business process. Even worse, the process contained in the system is often incorrect or incomplete.

5.1.2 Separation of management and execution
One important step towards achieving information systems that do fulfill the requirements listed above was their splitting into one subsystem that deals with the management of the business process (the "logistical system" or "management system") and one that supports the execution of tasks in a specific business process (the "application"; see figure 5.1). The management system deals with the logistical completion of cases, without actually performing tasks itself. It ensures that no steps are skipped, that they are carried out in the correct order, that tasks can be

![Figure 5.1](image)
The separation between logistics and execution
performed in parallel where possible, that the correct applications are called in to support a task, and so on. It also makes sure that staff are assigned, considers their absence, supports the separation of functions and authorization levels, and so on.

Apart from the structure of the business process, the management system actually has no application-specific characteristics. To differentiate between management and execution, in this book we use the principle that management may only consult the case attributes in order to wake routing decisions. We regard changing the case attributes as part of execution rather than management.

It is the task of the management system to bring the work (i.e., the work items) to the right person or application at the right time so that the tasks for a specific case can be carried out. The logistical management system interacts with the user, reacts to signals from its environment (for example, an incoming EDI message), or executes automatic or time-driven tasks. (In principle, a time-driven task also waits for a signal from the environment.) Once a supporting application for a particular step in the process has been defined, the management system starts this in the correct way. An application supports the user in performing the task. Management and applications communicate using case attributes. When an application is started, these can be passed on. When it closes again, any updated case attributes are passed back to the management level.

5.1.3 Advantages
Separating management from applications has a number of important advantages:

- It enables us to achieve uniform management functionality and to isolate this from the rest of the system. (Traditionally this functionality was spread throughout the information system.) This makes it possible to reuse the same functionality in more than one task.
- Applications no longer require any management functionality, and hence are simpler and completely independent of their context or place in the business process. This makes it possible to rearrange the business process at a later stage.
- The management layer makes it possible to integrate wide-ranging applications. In this way, it is even possible to integrate new applications with legacy systems.
• At the management level, the business process is identifiable and the state of a particular case within it is easy to establish. The process therefore is more traceable. Because it is clear at the management level which tasks have to be carried out, it is easy to determine who should be doing what for a particular case. The process execution is more manageable, with progress and bottlenecks more easy to check.

5.1.4 Workflow management software

Given that the process management functionality should, in principle, be widely applicable rather than intended for a specific application, it becomes attractive to use generic software: workflow management systems. These can interpret and apply the process structure and work allocation rules.

There is a large number of standard workflow management systems currently on the market. These vary widely in the functionality they offer. In this chapter, we shall try to indicate—in general terms—the functionality that one should or could expect from a workflow management system. In addition, we shall examine the technical aspects that are important in selecting and introducing such a system.

5.2 A Reference Model

As we saw in chapter 1, workflow management systems enable the "extraction" of process management from the application software. To a certain extent, we can compare such systems with a database management system. After all, database management systems make it possible to extract data management from the application software. Both types of systems support a piece of generic functionality. Because—unlike database management systems—workflow management systems have only been available for a short time, in many respects it is unclear which components are part of the systems' basic functionality. The technology is still young and not yet fully formed.

Moreover workflow management has many "faces." Workflow management systems may be implemented in order to achieve flexibility, system integration, process optimalization, organizational change, improved maintainability, evolutionary development, and so on. All this means that confusion may easily arise as to what actually can be expected from
The functionality of a workflow management system. This danger was recognized at an early stage by the Workflow Management Coalition (WFMC)—an organization whose role includes standardizing workflow management terminology and defining standards for the exchange of data between workflow management systems and applications. In 1996, the WFMC had already two hundred members (including many suppliers of workflow management products).

One of the many principles used by the WFMC is the so-called workflow reference model. This is a general description of the architecture of a workflow management system, in which the main components and the associated interfaces are summarized. Figure 5.2 illustrates the workflow reference model.

The model shows that the heart of a workflow system is the so-called workflow enactment service. This part of the system pumps—as it were—the cases through the organization. The enactment service ensures that the right activities are carried out in the right order and by the right people. In order to achieve this, use is made of process definitions and resource classifications produced by the so-called process definition tools. As well as illustrating the process and the organization, these tools fre-
Quently offer facilities for analysis techniques such as simulation. Work items are offered to the employees through workflow client applications. By selecting a work item, an employee can begin performing a specific task for a specific case. When carrying out a task it may be necessary to start an application. All the application software that can be started from the workflow system is known as invoked applications in the reference model. Workflow tracking, case control, and staff management are supported by the so-called administration and monitoring tools.

Five interfaces are also shown in figure 5.2. The WFMC is striving to standardize these. In creating an information system based upon a workflow management system, Interface 3 and Interface 4 are of particular significance. The former is associated with the control of applications from the workflow system, the latter with the exchange of cases (or parts of cases) between autonomous workflow systems. The other interfaces are mainly used by the workflow management system itself.

Figure 5.2 provides only a rough impression of the functionality of a workflow management system. We therefore shall further refine the definition of each component.

5.2.1 Workflow enactment service
The so-called workflow enactment service is the heart of a workflow system. This component creates new cases, generates work items based upon the process description, matches resources and work items, supports the performance of activities, and enables the recording of particular aspects of the workflow. For technical reasons, the enactment service may consist of several workflow engines. Their use can, for example, improve the scalability of the entire system. In an enactment service with more than one workflow engine, the work is distributed amongst them. This distribution may be based upon the characteristics of the case, the task, and/or the resource. In general, the user will not notice when a workflow system is using more than one engine.

Workflow engine A workflow engine provides those facilities which are required for the logistical completion of cases. In certain cases, several workflow engines operate alongside one another. Each then handles a portion of the cases and/or processes. The duties of a workflow engine include:
• creating new cases and removing completed ones;
• routing cases, using the interpretation of the appropriate process definition;
• managing case attributes;
• submitting work items to the correct resources (employees), based upon resource classification;
• managing and handling triggers;
• starting up application software during the execution of an activity;
• recording historical data;
• providing a summary of the workflow; and
• monitoring the consistency of the workflow.

The workflow engines are therefore the "core" of the workflow system, without which it would not operate.

5.2.2 Process definition tools
A workflow engine is based upon one or more workflow definitions. In chapters 2 and 3, we saw that the definition of a workflow is divided into two important parts: the process definition (chapter 2) and the resource classification (chapter 3). In the workflow reference model, the tools for constructing these are known as process definition tools. As well as tools for illustrating workflows, it is often also possible to make use of analysis tools. In chapter 4, we showed which analysis techniques are applicable in the context of workflow management. In principle, we thus can differentiate between three types of tools: (1) process definition tools, (2) resource classification tools, and (3) analysis tools. In a number of workflow management systems, these three tools are integrated into a single workflow definition and analysis tool. Please note that the term "process definition tools" used by the WFMC is slightly confusing, since it entails not only the tools for modeling process definitions, but also resource classification tools and analysis tools.

The process definition tool A process is specified using the process definition tool. Chapter 2 examined processes defined in terms of a Petri net. In many workflow management systems, however, processes are formulated in a different way. Nevertheless in most cases it is easy to map the used routing constructs onto Petri net elements. The expressive power of these alternative methods of modeling is typically weaker because
certain routing structures are excluded. For example, many workflow management systems abstract from the explicit modeling of states, and this does not allow for forms of routing such as the implicit OR-split to be modeled. The basic functionality of the process definition tool consists of the following elements:

• the ability to establish process definitions (name, description, date version, components, and so on);
• the ability to model sequential, parallel, selective, and iterative routing by means of such graphic components as the AND-split, AND-join, OR-split and OR-join;
• version management support (after all, there may be several versions of the same process);
• the definition of case attributes used in the process;
• task specification; and
• the checking of the (syntactical) correctness of a process definition and the tracing of any omissions or inconsistencies.

A number of characteristics need to be established for each task within a process. These determine the conditions under which that task may be carried out, and what operations should be performed. The following is established for each task:

• the name and description of the task;
• task information—in other words, any instructions and supporting information for the employee performing the task;
• the requirements with respect to the resource carrying out the task (for example, a specification of its role and organizational unit, or information about the separation of functions);
• the task's routing characteristics (AND-split, AND-join, OR-split, OR-join);
• the specification of any triggers required;
• instructions for the workflow engine (for example, priorities, case management, and resource management);
• the applications that may be started, plus the conditions and order in which this should be done;
• a specification of the case attributes that are used and adjusted by the application; and
• decision rules that determine the subsequent tasks based upon the case attributes, when there is an OR-split or mixed OR/AND-split.
The process established using the process definition tool is the crux of the workflow.

The resource classification tool As well as defining the process, the resources needed to carry out the workflow must be classified so that the tasks can be decoupled from specific employees. Most workflow management systems provide a resource classification tool in which the relationship between the various resource classes can be shown graphically. In doing so, the following items are established:

- a list of the resource classes, often subdivided into roles (based upon qualifications, functions, and skills) and organizational units (based upon arrangement into teams, branches, and/or departments);
- any specific characteristics of a resource class; and
- the relationship between the various resource classes (for example, a hierarchy of roles or organizational units).

The analysis tool Before a workflow that has been defined can go "into production," it is useful first to analyze it. Such analysis can encompass checking the semantic correctness of a process definition as well as performing a simulation in order to gain insight into the expected completion times for cases. In general we can state that the current generation of workflow management systems only offers limited analysis possibilities. In most systems it is therefore possible to define workflows that could have disastrous consequences if actually put into effect. However, as described in chapter 4, it is possible to apply advanced analysis techniques. Future workflow management systems therefore will offer more and more analysis possibilities.

5.2.3 Workflow client applications
Those employees who are only involved in the actual execution of a process will never use the process definition tools. The only contact they have with the workflow system is through the workflow client applications. Each employee has a worklist (also known as in-tray or in-basket) which forms part of the workflow client applications. The workflow engine uses this worklist to show which work items need to be carried out. By selecting a work item, an employee can begin performing a task for a specific case. In principle, therefore, every employee has a personal
worklist which shows all the work to be performed by him, or by his group. The worklist therefore forms the ultimate link between the work and the employee.

As shown in chapter 3, the allocation of work may be push or pull-driven. It is the former when the workflow engine allocates work items to individual employees. It is the latter when work items are allocated to groups of staff. This may result in a work item appearing in several worklists. The basic functionality that should be offered by a worklist handler encompasses the following:

- the presentation of the work items that may be performed by an employee;
- the provision of relevant properties of a work item, such as case and task information;
- the ability to sort and select, based upon these properties;
- the provision of state information pertaining to the state of the workflow engine;
- the starting of a task for a specific case when a work item is selected; and
- the ability to report the completion of an activity (i.e., a selected work item).

In addition, the worklist handler may allow for locking or passing on a work item. It must also be able to deal with system faults. Figure 5.3 shows a worklist handler of the CO$A$ workflow management system.

Most workflow management systems offer a so-called standard work-list handler. In some cases, though, it is necessary to create a customized worklist handler for a specific environment.

The standard worklist handler The standard worklist handler offers the functionality just described. Because it is not customized to suit a specific business situation, the functions available are generic. In many cases, however, it is possible to use parameters for the standard worklist handler. It may, for example, be possible to influence the layout and content of the window. Some standard worklist handlers have facilities for showing the (logistical) state of a case graphically.

The integrated worklist handler The only way in which a typical end user can access the workflow system is through the worklist handler. When such a system is supporting the work of, say, one hundred members
An example of a worklist handler (COSA, © Software-Ley)

of staff, the presentation of this component deserves particular attention. This may justify developing a customized worklist handler adapted to the specific business situation rather than using the standard one. This specific worklist handler would contain supporting facilities alongside the standard functionality described above. This is why it is referred to as an integrated worklist handler. It may, for example, use background data to provide additional support. Security and quality assurance considerations may also prompt the development of an integrated worklist handler. The same applies to the need for batch or chained processing of work items.

Batch processing is when an employee is able to perform a number of work items of the same type (in other words, repeat the same task) without switching back to the worklist handler. This enables her to carry out a particular task in routine several times in succession. Chained processing is when an employee is able to perform a number of successive
tasks for a specific case. In this way, she does not have to get used to a new case repeatedly. *Batch* and *chained processing* avoid continually and unnecessarily switching between the worklist handler and the applications. This can provide considerable returns in terms of efficiency.

### 5.2.4 Invoked applications

The performance of a task may result in the starting up of one or more applications. These do not form part of the workflow management system because they are associated with the actual performance of work, not to its logistical management. Such applications do belong to the workflow system, though. This, after all, encompasses the applications, configuration files, workflow management system, database, and so on. Applications are started by the workflow engine in order to perform a specific task. In doing so, information about the case may be submitted. The application may, for example, make use of a particular case-attribute value. The case's identification is frequently used to find the appropriate information in the database. Conversely, the application may change the case-attribute values. These modified attributes are often used to decide the routing of the case. In general, a clear distinction is drawn between *interactive* and *fully automatic applications*.

**Interactive application** An interactive application is always initiated as a result of the selection of a work item from the worklist handler. It may be a standard office tool such as a word processor or a spreadsheet, or a program developed especially for the business process (for example, an electronic form which needs to be completed).

**Fully automatic application** A fully automatic application requires no interaction with the user. It thus may be a part of a task that can be performed without a user intervening. One example could be a program which performs a complicated calculation (such as establishing the amount of an installment payment).

### 5.2.5 Other workflow enactment services

A workflow system may contain several workflow engines. These come under the same management and use the same workflow definitions. Such engines are said to belong to the same *workflow domain*. However
it is also possible to link several autonomous workflow systems with one another. In this way, cases (or parts of cases) can be transferred from one system to another. This means that the workflow enactment services of each system are linked. We refer to this as *workflow interoperability*. In the future, more and more workflow systems are expected to be linked. These may be in different branches of the same company or those of separate firms.

### 5.2.6 Administration and monitoring tools

The workflow enactment service ensures the processing of cases based upon workflow definitions. The supervision and operational management of these flows (including the resources) are done using *administration* and *monitoring tools*. These can be divided into those used for operational management of the workflows and those used for recording and reporting. In many workflow management systems they are integrated into a single tool.

**The operational management tool** Operational management covers all operations pertaining to the management of the workflow. So it is not possible to use the *operational management tool* to change the structure of a business process. We can subdivide the information related to operational management into that which is case related and that which is not (i.e., resource or system related). The operational management tool functions for resource-related information include:

- addition or removal of staff; and  
- input/revision of an employee's details (name, address, telephone number, role, organizational unit, authorization, and availability).

Additional operational management tool functions are:

- implementation of new workflow definitions; and  
- reconfiguration of the workflow system (setting of technical system parameters).

Note that an employee's individual details fall under operational management. The adjustment of employee availability information as a result of a revised schedule, holiday, or sick leave is one example of resource-related operational management. Functions for performing case-related operational management are also required:
• inspection of the logistical state of a case; and
• manipulation of the logistical state of a case due to problems and exceptional circumstances.

The operational management tool thus is also used to provide ad hoc solutions to problems resulting from system faults and bottlenecks in the process.

The recording and reporting tool Many aspects can be recorded and stored during the performance of a workflow. These are historical data which may be useful for management. For example, the following interesting performance indicators may be distilled from the data:

• average completion time for a case;
• average waiting time and processing time (possibly subdivided per task);
• percentage of cases completed within a fixed standard period; and
• average level of resource capacity utilization.

Note that in many situations not only the averages but also the variances of these performance indicators are of prime importance.

Information about the properties of completed workflows is crucial to management. Prompt warnings about bottlenecks and overcapacity can lead to the process being revised. The raw data is supplied by the workflow enactment service. It is then administered by the recording and reporting tool. This can, for example, decide at information should be stored. It also frequently offers reporting facilities. Some workflow management systems use predefined reports that are produced at regular intervals. Others offer an integrated report generator. This enables the user to define reports based upon the information recorded. And yet others deliberately do not provide reporting facilities. In this way, the recorded data can be found with the use of a standard database management system or a generic report generator. Often a huge amount of data needs to be translated in order to produce the information that is of interest to management. Clearly there is a link here with data mining, data warehousing, and OLAP (on-line analytical processing).

Figure 5.4 shows the relationship between the tools described. In fact, this illustrates a more detailed version of the workflow reference model given in figure 5.2. It does not, though, state that the analysis tool and
Figure 5.4
The various components of a workflow system

the recording and reporting tool often make use of one another's information. For example, historical data can be used in analyzing a workflow (through, say, simulation). Analytical results can also be used in dedicated searches for useful management information.

5.2.7 Roles of people involved
Figure 5.4 clearly shows that a workflow system is constructed from many components that are used by a wide range of people. In theory, there are four types of users:

• The Workflow Designer. The workflow designer uses the process definition tools (in other words, the process definition tool, the resource classification tool, and the analysis tools). This designer works on the structure of the workflow.
• **The Administrator.** The administrator uses the operational management tool. His typical activities include adding employees, issuing and withdrawing authorizations, implementing new processes, monitoring workflows, and solving problems and bottlenecks.

• **The Process Analyst.** The process analyst uses the recording and reporting tool to inform the management about the performance of the workflows. By aggregating detailed data into performance indicators, it is possible to provide insight into the operation of the business processes that are supported by the workflow management system.

• **The Employee.** The execution of work is carried out by employees. In this book, they are also referred to as resources. Such resources are the scarce means of production which need to be employed in the best way possible.

As well as the four types of users, other people are often involved in the structuring, management, and performance of the workflows. The users of the workflow management system are usually led by a manager. New and/or revised workflows often require new or updated applications. Information requirements may also be changed by the introduction of a new process. This is why database designers/programmers and application designers/programmers are also involved in the (re) structuring of a workflow. Figure 5.5 shows the various types of people involved in workflow design, implementation, and enactment.

It goes without saying that, in practice, the distinction between people and roles is not always as clear-cut as shown in figure 5.5. The process

![Figure 5.5](image_url)

The users of a workflow management system
analyst may also be a manager, an employee also an Administrator—and there may be several types of administrators. In chapter 6, we shall examine in more detail the various types of people involved in implementing and managing workflow systems.

5.3 Storage and Exchange of Data

A workflow system consists of a large number of components. For the whole system to operate properly, these components must exchange information with one another. Furthermore it is important that different sorts of data are stored. Using figure 5.4 we shall show which data is administered within the workflow system. We shall then examine the links between the various components.

5.3.1 Data in a workflow system

Figure 5.4 shows which data is of significance to the workflow system. In most cases the workflow management system and the applications make use of the same database system. The workflow system thus "contracts out" data administration to a database management system. The following data sets are involved:

1. Process definitions. The definitions of processes and tasks. The name, description, routing, tasks, and conditions of each process are recorded. For each task, its name, description, decision rules, content, and allocation rules are recorded.
2. Resource classifications. The structuring of the various types of resources. As well as a list of resource classes (roles or organizational units), the relationships between them are recorded.
3. Analysis data. The results of any analyses carried out. In the case of simulations, for example, subrun results. (A simulation also sometimes makes use of historical data.)
4. Operational management data. The data that are important to the administrator of the workflow system. For example, information about the technical configuration of the system (system parameters), information about staff, and case-related data.
5. Historical data. The data that are stored in order to be able to retrace the progress of an individual case, trace the cause of a problem, or assess the performance of the business process.
6. **Application data.** The data that can be accessed by an application but not by the workflow management system. There are two types of application data: case data and master data. Case data are directly related to individual cases; master data are not. The latter includes general information about customers and suppliers.

7. **Internal data.** All the data that are maintained by the workflow management system but are not directly related to the workflow as such. For example, information about worklists that are active, the state of each engine, and network addresses. Unlike the operational management data, the internal data are technical in nature and therefore are only accessed by the enactment service.

8. **Logistical management data.** The state of each workflow is embedded in the logistical management data, which encompass information about case states (including case attributes), the state of each resource, and the triggers available. It is preferable that these are accessible only by the workflow engine. However, it is for technical reasons sometimes unavoidable that these are also consulted, and even revised, by external applications.

### 5.3.2 Interfacing problems

A workflow system consists of a large number of components. Some of these are the workflow management system tools themselves, while others are the applications used when carrying out the actual tasks. In order for these components to work together, they must exchange information. Agreements have therefore been reached within the WFMC about the standardization of interfaces between the various components. As shown in figure 5.6, the WFMC recognizes five such interfaces.

The objective of interface standardization is threefold. First, generally accepted standards will improve the exchange of data between (parts of) workflow management systems. Second, it will become possible to create links between different manufacturers' enactment servers in a simple way. Finally, the standards will enable the development of applications that are entirely independent of the chosen workflow management system.

A number of interfaces are currently achieved using files or databases. For example, in figure 5.4, we have assumed that Interface 1 and Interface 5 are realized using a database. Within the WFMC, however, it is assumed that every interface will be achieved using a so-called **application programming interface (API)**. In the context of workflow management, the term WAPI (workflow application programming interface) is
Figure 5.6
The interfaces between the various elements (© WEMC)
also used. An API is a group of services that are offered to a client via a server. These services can best be compared with procedure calls in a conventional programming language. The word client can refer to an application. An operating system such as UNIX is an example of a server. We can consider the copying of a file as a service offered by UNIX via an API (cp). In the specific case of workflows (WAPI), the workflow enactment service acts as the server and the tools and applications as clients. To provide an impression of the WAPIs recognized by the WFMC, we shall briefly describe the content of each interface:

1. **Interface 1 (process definition tools).** Interface 1 provides the link between the tools designed for creating and modifying the workflow definitions (process definition tools) and the workflow enactment service. This WAPI contains functions for opening and closing a connection (connect/disconnect), obtaining a summary of the workflow definitions (process definitions and resource classifications), and opening, creating, and saving a process definition.

2. **Interface 2 (workflow client applications).** The second interface is dedicated to communication between the worklist handler and the enactment service. The WAPI that enables this supports, among others, the following functions: opening and closing of a connection, production of case and work item state summaries, generation of new cases, and the beginning, interruption, and completion of activities.

3. **Interface 3 (invoked applications).** An application is opened from the workflow management system through Interface 3. Figure 5.6 suggests that every application is opened directly from the workflow enactment service, but this is not always the case. An interactive application such as a word processor will generally be opened from the worklist handler.

4. **Interface 4 (other workflow enactment services).** Interface 4 enables the exchange of work between several autonomous workflow systems (for example, case transfers and the outsourcing of work items). This WAPI thus facilitates workflow interoperability.

5. **Interface 5 (administration and monitoring tools).** Interface 5 is concerned with the link between administration and monitoring tools and the workflow enactment service. It is subdivided into two parts: workflow system management functions and workflow tracking functions. The former could include the addition of an employee, the permission of authorization, and the execution of a process definition. To track a workflow, the enactment service records a wide variety of events in a logfile. Specific questions about this historical data can be posed via Interface 5. These could cover waiting times, completion times, processing times, routing, and staff utilization.
The WFMC is still working on standardizing the WAPIs. For example, little progress has been made thus far in agreeing on standards for Interfaces 3 and 5. Nevertheless the discussion about the five interfaces provides a good impression of the functionality desired of a workflow management system.

For those involved in the introduction of a workflow management system, Interface 3 is of particular importance. Interface 4 only becomes significant when one wishes to link more than one workflow system. Interface 2 enters the picture when the standard worklist handler is no longer adequate and an integrated application needs to be developed. Interface 5 becomes significant when one wishes to compile management information from the events recorded by the enactment service. In practice, Interfaces 3 and 4 appear to cause most problems. We therefore shall consider their potential difficulties in more detail.

Figure 5.7 shows diagrammatically how an application can be started (Interface 3). This may be done by an engine and/or from a worklist handler. An application is called to perform a task. Say the engine begins the performance of a task and so starts up an application. This application probably will modify application data in the database. If the workflow engine does not become accessible following the execution of the application due to a system error, then the engine and the application will be "out of synch." Once the system has been corrected, the engine will have no choice but to rollback the task. After all, it has no way of
knowing that the application has completed the task successfully, and any changes in the case attributes have not been passed on. This results in the logistical data (case state) and application data no longer matching. Disastrous consequences may follow. Consider, for example, a payment by a bank: if the application has made the payment but the workflow management system is not aware of this because of a fault, then the same payment may be made again.

Similar problems may occur when an application is opened from the worklist handler. Assume that an error in the worklist handler occurs while the application is running. Again the workflow system and the application become "out of synch." The fact that the engine, database, worklist handler, and application can all operate on different systems only makes these problems worse. In a client/server environment, for example, the worklist handler and part of the applications run locally (client), but the rest operates centrally (server). To solve such problems effectively, it is vital that the engine, the database, the worklist handler, and the application all regard a task (or a part of a task) as a common logical unit of work (LUW). This means that the so-called ACID properties (atomicity, consistency, isolation, and durability) apply:

- **Atomicity.** A task either is completed successfully in full (commit) or restarts from the very beginning (rollback).
- **Consistency.** The result of an activity (in other words, the performance of a task) leads to a consistent state.
- **Isolation.** If several tasks are carried out simultaneously, the result is the same as if they had been carried out entirely separately. In other words, tasks performed at the same time should not influence one another. This property is also referred to as "serializability."
- **Durability.** Once a task is successfully completed, the result must be saved. A task therefore must be completed with a commit that ensures that the result cannot be lost.

Within classic transaction processing environments like those we encounter in the financial world, we frequently have to "pass the ACID test." In practice, though, with the current generation of workflow management systems, it appears not to be easy to address the ACID properties in full. This aspect therefore deserves to be taken fully into account at an early stage.

We encounter similar problems when linking two or more workflow systems (Interface 4). In addition, in most workflow management systems
it is not always entirely clear what the state of a case is. In terms of Petri nets, the state of a case corresponds with the distribution of tokens amongst places (conditions) and the values of case attributes. The transfer of a case between two workflow systems based upon Petri nets therefore is equivalent to transferring tokens and case attributes. In many other workflow systems the situation is not so simple, because they often abstract from the state of a case at the conceptual level. (The places are omitted from the definition of the process.) In such cases, complicated "translation" work is required to transfer a case from one system to another. Note that, in addition to transferring cases, the outsourcing of work items and the generation of new cases in a different system also fall within the scope of workflow interoperability.

5.3.3 Interoperability standards

The presentation in this chapter is based on the reference model of the WFMC. This model was chosen as a starting point since it provides a nice introduction to workflow technology. Many authors have criticized the reference model as being too naive or emphasizing the wrong issues. In this chapter we will not compare the reference model to alternative architectures: These more technical discussions are outside the scope of this book. However, we will point out recent efforts to resolve the interoperability problems identified in this chapter.

In the last couple of years several interoperability standards, that is, specifications for the exchanging information between workflow products, have been proposed. We can classify these interoperability specifications into two categories: specifications for workflow modeling and workflow description (i.e., design-time) and specifications for run-time interoperability.

The first category corresponds to Interface 1 of the reference model of the WFMC. The WFMC's process definition language (WPDL) falls into this category. Another example is PIF (process interchange format). PIF is an interchange format designed to help automatically exchange process descriptions among a wide variety of process tools such as process modelers, workflow systems, process repositories, etc. These tools can interoperate by translating their native process description format to PIF, and vice versa. In this way, process descriptions can be exchanged automatically without using different translators for each pair of systems. If a translation to or from PIF cannot be achieved automatically, human
efforts are needed. The PIF format did not gain sufficient momentum to become an industry standard. However, many of the ideas have been adopted by a new initiative: the process specification language (PSL). PSL is promoted by NIST (U.S. National Institute of Standards and Technology) and has a scope which is much broader than the WPDL of the WFMC. There are several even more general standards emphasizing different aspects, that is, the standardization efforts in the context of UML (statechart diagrams, sequence diagrams, collaboration diagrams, and activity diagrams), the ISO standard for (high-level) Petri nets (ISO/IEC JTC1/SC7/WG11), and the well-known IDEF0 standard (also supported by NIST). These standardization efforts are relevant but clearly provide no solution for today's design-time interoperability problems. This is a result of the absence of a common conceptual or formal core model, as was mentioned before.

The second category of interoperability specifications is concerned with run-time interoperability. This category corresponds to Interface 2, Interface 3, and Interface 4, with a focus on Interface 4. The focal point is on the support of exchanging process enactment information at run-time. Clearly, Interface 4 is of the utmost significance when exchanging enactment information between systems of different vendors. The most notable initiatives with respect to run-time interoperability are the Interoperability Specification of the WFMC, SWAP, WF-XML, and OMG's jointFlow. Already in 1996, the WFMC released the Interoperability Abstract Specification (WFMC-TC-1012). This was followed by the so-called Interoperability Internet e-mail MIME Binding (WFMC-TC-1018). Recently (May 2000), the WFMC released the so-called Interoperability Wf-XML Binding (WFMC-TC-1023). The latter describes a realization of the Interoperability Abstract Specification using XML and is based on SWAP. SWAP (Simple Workflow Access Protocol) is an Internet-based standard and supported by multiple workflow vendors. SWAP heavily uses the HTTP protocol and can be used to control and monitor workflow processes. OMG's jointFlow is an initiative based on the CORBA architecture and also uses the Interoperability Abstract Specification of the WFMC as a starting point. The jointFlow standard is formed by a set of IDL specifications. The standards concerned with run-time interoperability are very relevant for the realization of workflow systems. In the context of electronic commerce, these standards will become even more
Figure 5.8
A summary of the technical components

important. Unfortunately, the standards are at a rather technical level and do not really deal with issues at a business level. It is possible to connect systems of different vendors using for example Wf-XML. However, this does not imply that the process is executed as intended.

5.4 Required Technical Infrastructure

In achieving a functional workflow system, it is not sufficient simply to purchase a workflow management system. As shown in figure 5.8, this is only one of the components required.

The successful introduction of a workflow system requires a suitable technical infrastructure. Most operate within a client/server environment. Such an environment typically consists of a central server operating in Windows NT/2000 or UNIX and a number of clients using MS-DOS/Windows 3.1, OS/2 or Windows 95/98/2000. As we have already seen in figure 5.7, the workflow engine operates on the server side. The worklist handler, and hence the user interface, operates on the client side. The applications may operate on either side. The database of management and application data is administered by the server. Without becoming mired in a technical explanation, we shall briefly consider the main components:

1. **Hardware.** The server is usually a powerful microcomputer, or a mini or mainframe computer. Reduced instruction set computers (RISCs) are often used. Clients are generally choosing complex instruction set com-
puters (CISCs): for example, personal computers (PCs) based upon Intel 80x86 processors. The server is linked to the clients using coax, (un)shielded twisted pair or fiber-optic cable. Bridges, routers, hubs, and/or gateways are also required when building large networks.

2. **Operating system.** The operating system of the server should allow for multiple users and multitasking. One obvious choice is UNIX; other possibilities are OS/2, Windows NT/2000, or Linux. Mainframes are seldom used for workflow management. Operating systems like VMS, MVS, and AS400 are also rarely supported by the current generation of workflow management systems. The client's operating system is usually Windows 95/98/2000. However, it could also use UNIX, OS/2, or Linux. One characteristic of modern operating systems is that they support user interfacing.

3. **Network software.** The network plays a crucial role in the operation of a workflow system. It links the clients with the server. Common choices of network technology are the Ethernet and the Token Ring protocol. The communications software uses such a protocol to exchange messages. TCP/IP (Transmission Control Protocol/Internet Protocol) is currently the most widely-used standard in client/server environments. Other possibilities are NetWare, SNA, OSI, and AppleTalk.

4. **Database management system.** Many information systems are constructed around a database system. In a workflow system, too, the database plays a major role. Usually the applications and the workflow management system use the same database system. This means that the workflow management system must be able to make use of a database management system that has already been chosen. Most workflow management systems therefore support the most common relational database management systems such as Oracle, Sybase, and SQLServer. Using ODBC (open database connectivity) it is, in theory, even possible to make the workflow management system independent of the underlying database management system. However, the selection of an incompatible combination can result in poor performance by the entire workflow system.

5. **Applications.** The applications support the performance of tasks. They may be either standard software packages, such as a word processor or a spreadsheet, or customized software written in a script language, a third-generation language (such as C++ or Java), or a fourth-generation one (like Powerbuilder or Oracle Designer/2000). Various mechanisms are conceivable for starting up an application. Firstly, a command line can be used (in other words, it is started directly from the operating system). The case attributes can be exchanged through a WAPI or the database. The drawback to this is that a new program must be
Functions and Architecture of Workflow Systems

started for each activity. It therefore is sometimes better to start the application only once. In such a case, the application is not closed when an activity is completed. So starting it a second, third, or fourth time is no longer necessary. In Windows, for example, DDE (Dynamic Data Exchange) is used to achieve this.

6. **Workflow management system.** The workflow management system has to deal with each of the components listed above. It must be able to exchange information with the applications and the database system. Moreover, it must be able to cope efficiently with the available processing and network capacity.

The above shows that technical as well as functional aspects need to be taken into account when selecting a workflow management system. Such a system uses the hardware, operating system, network software, database management system, and applications already in place. It therefore is vital that the chosen workflow management system suits those components. A poor combination can result in an unreliable system with long response time and a low processing speed.

### 5.5 Current Generation of Workflow Products

Today, many workflow management systems are available. Figure 5.9 lists some of them. This list is just a snapshot: It is far from complete and the support for some of the products listed has been discontinued. The number of suppliers offering workflow management software is estimated at two hundred—which indicates that such systems are expected to play a major role in the near future. Besides the specialized workflow management systems, most ERP-systems such as SAP, Baan, and JD Edwards have a workflow engine incorporated. In most cases these workflow engines cannot be used as standalone workflow management systems.

The information in this chapter is based upon the situation in early 2000. Due to the rapid pace of developments in the workflow market, this picture is likely to change completely within a few years. The rest of this book is, however, less time-dependent and will therefore remain current for many years to come.

Despite the large number of suppliers, some of which are listed in figure 5.9, the number of workflow systems actually in production is
<table>
<thead>
<tr>
<th>ActionWorkflow</th>
<th>Action Technologies Inc.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computron Workflow</td>
<td>Computron</td>
</tr>
<tr>
<td>COSA</td>
<td>CSE</td>
</tr>
<tr>
<td>CSE/Workflow</td>
<td>Universal Systems Inc.</td>
</tr>
<tr>
<td>Documetrix Workflow</td>
<td>BancTec-Plexus</td>
</tr>
<tr>
<td>FloWare</td>
<td>PowerCerv</td>
</tr>
<tr>
<td>FLOWBuilder</td>
<td>IBM</td>
</tr>
<tr>
<td>FlowMark/MQ Series Workflow</td>
<td>Delrina</td>
</tr>
<tr>
<td>FormFlow</td>
<td>Empirica</td>
</tr>
<tr>
<td>HICOS</td>
<td>TIBCO/InConcert</td>
</tr>
<tr>
<td>InConcert</td>
<td>Promatis</td>
</tr>
<tr>
<td>Income</td>
<td>JetForm Corporation</td>
</tr>
<tr>
<td>JetForm Server</td>
<td>UES Inc.</td>
</tr>
<tr>
<td>KI Shell</td>
<td>I. Levy &amp; Associates</td>
</tr>
<tr>
<td>NAVIGATOR 2000/Workflow</td>
<td>Wang</td>
</tr>
<tr>
<td>Open Workflow</td>
<td>SNS Systems</td>
</tr>
<tr>
<td>OPEN IMAGE</td>
<td>Optika Imaging Systems Inc.</td>
</tr>
<tr>
<td>PowerFlow</td>
<td>Cap Gemini Innovation</td>
</tr>
<tr>
<td>Process Weaver</td>
<td>SAPAG</td>
</tr>
<tr>
<td>SAP Business Workflow</td>
<td>Staffware</td>
</tr>
<tr>
<td>Staffware</td>
<td>TeamWARE</td>
</tr>
<tr>
<td>Ultimus</td>
<td>Ultimus</td>
</tr>
<tr>
<td>Verve</td>
<td>Verve Inc.</td>
</tr>
<tr>
<td>ViewStar</td>
<td>ViewStar</td>
</tr>
<tr>
<td>Visual Workflow</td>
<td>FileNet Corp.</td>
</tr>
<tr>
<td>WebFlow</td>
<td>Cap Gemini Innovation</td>
</tr>
<tr>
<td>Workflow Factory</td>
<td>Delphi Consulting Group</td>
</tr>
<tr>
<td>WorkFLOW SQL</td>
<td>Optical Image Technology Inc.</td>
</tr>
<tr>
<td>WorkParty</td>
<td>Siemens Nixdorf IS-AG</td>
</tr>
<tr>
<td>WorkVision</td>
<td>IA Corporation</td>
</tr>
</tbody>
</table>

Figure 5.9
A number of workflow management systems and their suppliers

relatively limited. There are several reasons for this. First, the technology is quite new, so systems developers often are insufficiently aware of the possibilities offered by a workflow management system. Also many workflow management systems still are not fully developed, resulting in limited functionality and unsatisfactory reliability. And it is currently not easy to opt for a specific workflow management system. The large number of systems available and the high degree of uncertainty about the future make the choice even more difficult. Finally, despite the efforts of the WFMC, standards with respect of functionality and system linking are lacking. For example, many workflow management systems use an ad hoc drawing technique to specify processes. One of the drawbacks of this is that it is difficult to exchange process descriptions between differ-
functions and architecture of workflow systems. (A conceptual standard based upon Petri nets would make a significant contribution in this respect.) Despite these obstacles, the importance of workflow management will only increase in the future.

In order to gain an impression of the current generation of workflow management systems, we shall briefly examine three products: Staffware® (Staffware Plc), COSA® (Ley GmbH), and ActionWorkflow® (Action Technologies Inc.). Staffware is one of the leading workflow products with an estimated market share of twenty-five percent. Therefore it serves as a nice illustration of the capabilities of today’s workflow management systems. The latter two products have been chosen because they represent extremes in the broad spectrum of workflow management systems. COSA is a robust product with extensive possibilities for managing complex business processes. It also closely shadows the process modeling technique used in this book. ActionWorkflow represents an entirely different approach, in which the emphasis is placed upon coordinating the parties involved rather than managing the process. Staffware will be discussed in some detail. The other two are discussed only briefly. We will also present some tools for workflow analysis and BPR and mention some criteria for selecting a workflow management system.

5.5.1 Staffware
Staffware® is one of the most widespread workflow management systems in the world. In 1998, it was estimated by Gartner Group that Staffware has twenty-five percent of the global market. Staffware Plc, the company that develops and distributes Staffware, is headquartered in Maidenhead, U.K. In this section we describe the current version of Staffware: Staffware 2000. Staffware 2000, the successor of Staffware 97, was launched at the end of 1999.

Staffware consists of the following components:

1. **Graphical Workflow Definer (GWD).** The GWD is the process definition tool of Staffware. It does not support any form of analysis.
2. **Graphical Form Designer (GFD).** The GFD is used to define the interface that is presented to the end-user or, in case of an automatic task, the interface that is presented to the external application.
3. **Work Queue Manager (WQM).** The WQM is the client tool of Staffware which is used to offer work to end-users.
4. **Staffware Server (SS).** The server component of Staffware takes care of the run-time enactment of the workflow.

5. **Staffware Administration Managers (SAM).** The SAM consists of a set of tools to support the workflow administrators. The following tools are included: user manager, backup manager, table manager, case manager, list manager, network manager, and sysinfo.

6. **Audit Trail (AT).** The AT facility is used to monitor the execution of individual cases.

The Staffware components can be mapped onto the reference model of the WFMC quite easily: GWD and GFD correspond to the process definition tools (Interface 1), WQM corresponds to the workflow client applications (Interface 2), SAM and AT correspond to the administration and monitoring tools (Interface 5), and SS provides the workflow enactment service of Staffware.

Figure 5.10 shows a screenshot of the GWD. The modeling language used is specific for Staffware. The tasks are called **steps.** There are several kinds of steps: automatic steps (offered to an application instead of an end-user), normal steps (executed by an end-user), and event steps (triggered by some external event). The semantics of a step are **OR-join/AND-split;** that is, a step becomes enabled if one of the preceding steps is completed and the completion of step will trigger all subsequent steps. Since the OR-join/AND-split semantics is fixed, two additional building blocks are needed: the wait step and the condition. The wait step can be used to synchronize flows and has AND-join/AND-split semantics. To model choices, that is, OR-splits, the condition building block can be used. Staffware only allows for binary choices, that is, just two possible outcomes (e.g., YES and NO). Staffware processes always start with a start step that is denoted by a symbol representing a traffic light. Termination in Staffware is implicit; it is possible to start multiple parallel threads that end concurrently. Therefore there is no need to have one sink node representing the completion of a case. The end of a thread is denoted by a stop symbol. Conditions are modeled by diamond-shaped symbols. Wait steps are modeled by symbols in the shape of a sand timer. The basic semantics of a step, a condition, and a wait are shown in figure 5.11.

The translation shown in figure 5.11 does not consider two additional features available for steps. First of all, it is possible to withdraw steps.
Figure 5.10
The graphical Workflow Definer (GWD): The design tool of Staffware

Figure 5.11
The semantics of some of the Staffware constructs (left) expressed in Petri nets (right)
Second, it is possible to model a time-out, that is, a step triggering other steps if it is not executed within a given time period.

Figure 5.12 shows the process of handling insurance claims used in chapter 2 modeled with the Staffware GWD. Figures 5.11 and 5.12 show that the modeling language used by Staffware is quite similar to the technique used throughout this book: concepts such as AND/OR-split/join play an important role in both types of models. Nevertheless there are some subtle, but relevant, differences. One of the core differences is the fact that the notion of states, that is, a concept similar to places, is not supported by Staffware. As a result, some models may appear to be more straightforward in Staffware (e.g., a simple sequential process). However, other models become larger as a result of the binary choice and the need to introduce wait steps for synchronization purposes. In fact, several constructs that can be modeled in terms of Petri nets cannot be modeled in Staffware, such as implicit choices, milestones, and other non-free-choice constructs. The only way to support these constructs is to encode the functionality in an external application or accept different semantics.

Staffware does not offer a tool for organizational modeling. Instead Staffware uses the concept of the work queue. A work queue can be compared to a resource class. Every queue is associated with a group of users. A user can be a member of many work queues and a work queue can be associated with many users. Each user sees the work queues for which she is a member of the associated group. Work items can be put into one or more work queues. If a work item is put into a work queue, one of the associated members has to execute the work item. When a user wants to process a work item, she selects it from its queue. While the user is processing the work item, the work item remains locked for all other members of the group. After processing, the user can either release the item (i.e., tell the system the work item is done), or put it back into the queue.

Figure 5.13 shows the WQM of Staffware. This tool is used to offer the work to end-users. On the lefthand side the work queues are shown. Note that each user has one personal work queue and several group queues. Figure 5.13 shows four group queues. On the righthand side some of the work items are shown. By selecting a specific queue, the user can see all work items corresponding to this queue. In figure 5.13 there are three work times corresponding to the work queue IC CD Employee.
Figure 5.12
A Staffware process for handling insurance claims
Figure 5.13
The Work Queue Manager (WQM) of Staffware
Figure 5.14 shows some other tools offered by Staffware 2000. The Audit Trail tool (top right) shows a trace of all occurrences for a given case or process. The User Manager (bottom) is used to maintain a list of end-users, privileges, queue membership, etc. The User Manager is just one of the Staffware Administration Managers (SAM) tools.

This concludes our introduction to Staffware 2000. It nicely illustrates the features of the current generation of workflow management systems. The description of the two other workflow management systems (COSA and ActionWorkflow) will be less elaborate.
COSA® (COMputerunterstützte SAchbearbeitung) is produced by Software-Ley GmbH. It is a workflow management system based upon Petri nets. COSA can be described as a traditional workflow management system that closely follows the WFMC reference model. It is also characterized by very extensive functionality and a somewhat dated user interface. The figures shown in this section exhibit COSA 1.4. The user interface of COSA 2.0 and the recently released COSA 3.0 looks quite different but—in essence—offers the same functionality. COSA consists of the following components:

1. **COSA Network Editor (CONE).** CONE is a process definition tool for defining and revising processes. As shown in figure 5.15, Petri nets are used to illustrate processes.

2. **COSA User Editor (COUE).** COUE is a resource classification tool for defining roles and organizational units. Figure 5.16 shows how resource classes can be structured hierarchically.

3. **COSA MemoBox (COMB).** COMB is a standard worklist handler for offering and starting work items (see figure 5.3). Every employee is provided with her own worklist handler.

4. **COSA Networkstate Displayer (COND).** COND is a graphic tool for presenting the state of a case. Because an employee can see the state of a case, she is aware of the business process.

**Figure 5.15**
A COSA process definition produced with CONE
5. **COSA Runtime Server (CORS).** The COSA Runtime Server is a workflow enactment service which consists of one or more engines.

6. **COSA Simulator (COSI).** COSA offers a primitive tool for simulating business processes. There is also a link available between COSA and the analysis tool ExSpect.

7. **COSA Administrator (COAD).** COAD is used to manage the work flows. COSA does not offer a recording and reporting tool. However standard reporting tools (such as Management Information Systems, OLAP, and Extraction tools) can read and process the information required from the COSA database.

COSA's architecture can easily be mapped onto the WFMC reference model (see figure 5.2). CONE, COUE, and COSI form the process definition tools (Interface 1). COMB and COAD respectively correspond with the workflow client applications (Interface 2) and the administration and monitoring tools (Interface 5). COND can be regarded as supplementing COMB.

COSA supports many technical platforms, including UNIX, Windows NT/2000, and OS/2 on the server side and OS/2, Windows NT/2000, Windows 3.1, Windows 95/98/2000, and UNIX on the client side. The following database management systems are supported: Oracle, Inforix, Sybase, Ingres, and DB2. It is also possible to communicate with running workflows via the Internet using COSA Portal; that is, it is possible to access the memobox functionality via a web browser.
5.5.3 Action Workflow

Action Workflow® is produced by Action Technologies Inc., and focuses upon supporting processes in which communication between people and/or parties plays a major role. In this sense, ActionWorkflow is very different from more traditional workflow management systems like COSA and Staffware. Unlike COSA and Staffware, which concentrate upon the process, ActionWorkflow centers on coordination. Action-Workflow uses so-called Business Process Maps (BPM). These are constructed from a number of workflows (see figure 5.17). Each workflow corresponds with a transaction that passes through the following stages: (1) preparation, (2) negotiation, (3) performance, and (4) completion. Transitions between these stages take place using so-called speech acts (communication between the people/parties involved in the transaction). Workflows can be linked with one another to illustrate the connections between the transactions. In this way, refinements and various types of routing can be shown. In the BPM illustrated in figure 5.17, workflows D and E are carried out in parallel. Workflow C is performed after workflow B.
In this section we discuss the functionality of ActionWorkflow 3.0. This is not the current workflow product of Action Technologies Inc. The focus of Action Technologies Inc. shifted from pure workflow management to complete business solutions. However, their latest product, called ActionWorks Metro (a so-called "e-process application platform"), includes the functionality of ActionWorkflow 3.0.

ActionWorkflow 3.0, also known as the ActionWorkflow Enterprise Series, consists of the following components:

2. **ActionWorkflow Process Manager.** The Process Manager is at the heart of ActionWorkflow. It is both a workflow engine and a tool for managing the workflow. In addition it offers advanced possibilities for analyzing workflows which are in progress.

3. **Action DocRoute.** DocRoute is based upon the Process Manager and offers the ability to integrate document management and imaging applications seamlessly.

4. **Action Metro.** Action Metro offers the opportunity to create workflow systems which make use of the Internet. Web browsers such as Netscape Navigator and Microsoft Internet Explorer hence can be used as worklist handlers.

We can also illustrate the ActionWorkflow components using the WFMC reference model. ActionWorkflow Process Builder is the only process definition tool (Interface 1). ActionWorkflow Process Manager corresponds with the workflow enactment service, the administration and monitoring tools (Interface 5) and part of the workflow client applications (Interface 2). Action DocRoute is difficult to place in the reference model. Action Metro can be treated as an alternative to Interface 2; a Web browser acts as the Workflow Client Application.

ActionWorkflow is only available for a limited range of platforms. ActionWorkflow 3.0 is only available for Windows NT/2000 on the server side. The Process Builder also operates under Windows 95/98/2000. Through the use of the Internet, the client software is suitable for almost every system. Data management makes use of Microsoft SQLServer.

The above shows that COSA (or Staffware) and ActionWorkflow are two very different workflow management systems. COSA is traditional and thorough, enabling the support of most routine production processes within administrative organizations. ActionWorkflow differs in many respects from standard workflow management systems, and appears to be best suited to supporting processes in which coordination is crucial.

### 5.5.4 Analysis tools

As was pointed out in the previous chapter, there are several techniques for analyzing workflow systems. Unfortunately contemporary workflow management systems hardly support any form of analysis. In chapter 4 we differentiated between qualitative analysis (concerned with the logical correctness) and quantitative analysis (concerned with the performance
and capacity requirements). Only a few workflow tools focus on qualitative analysis. Most of the workflow management systems have only trivial correctness checks, such as: is the workflow graph connected? More advanced checks like the absence of deadlocks, guaranteed termination, and proper termination are not supported. A few research tools have been developed to tackle the problem of qualitative analysis. Most notable are Woflan (SMIS/I&T, Eindhoven University of Technology, The Netherlands) and FlowMake (DSTC Pty Ltd, The University of Queensland, Australia). Both tools are capable of analyzing properties similar to the soundness property defined in chapter 4. Many of the workflow management systems available today support some export facility to simulation tools. This export facility is used to analyze the quantitative aspects of a workflow process. An example is the link between Staffware and Stuctware/BusinessSpecs (IvyTeam, Zug, Switzerland). Another example is the link between COSA and ExSpect (Deloitte & Touche Bakkenist, The Netherlands).

To illustrate the functionality of these analysis tools we briefly describe two products: Woflan and ExSpect.

**Woflan** Woflan (WOrkFLow Analyzer) is a tool that analyzes workflow process definitions specified in terms of Petri nets. It has been designed to verify process definitions that are downloaded from a workflow management system such as Staffware and COSA. As indicated in chapter 4, there is a clear need for such a verification tool. Today's workflow management systems do not verify the correctness of workflow process definitions. Therefore errors made at design time such as deadlocks and livelocks may remain undetected. This means that an erroneous workflow may go into production, thus causing dramatic problems for the organization. To avoid these costly problems, it is important to verify the correctness of a workflow process definition before it becomes operational.

The development of the tool Woflan started at the end of 1996, and the first version was released in 1997. Basically, Woflan takes a workflow process definition imported from some workflow product, translates it into a Petri net, and tells whether or not the net is a sound workflow net. Furthermore using some standard Petri-net analysis techniques as well as those tailored to workflow nets, the tool provides diagnostic
information about the net in case it is not a sound workflow net. Version 2.0 of Woflan has an import facility for COSA, Staffware, METEOR, and Protos. Figure 5.19 shows a screenshot of Woflan. A trial version of Woflan can be downloaded from http://www.tm.tue.nl/it/woflan.

ExSpect *ExSpect* (Executable Simulation Tool) is a full-fledged simulation tool based on Petri nets. The development of ExSpect started in 1988 at Eindhoven University of Technology as a research prototype. In the mid-1990s the development moved to the Dutch consultancy company Bakkenist. At the moment ExSpect is supported by Deloitte & Touche Bakkenist, The Netherlands. The application of ExSpect is not limited to workflow analysis. ExSpect can also be used to simulate production processes, transportation networks, software components, embedded systems, etc. In fact, ExSpect can be used to prototype simple systems and can interact with run-time systems via the Microsoft COM standard. However, for this book, the link between ExSpect and several workflow products is most relevant. ExSpect can download workflow processes from workflow management systems such as COSA and BPR tools such as Protos. Figure 5.20 shows a screenshot of ExSpect. The screenshot shows that ExSpect supports graphical animation of the workflow processes. In addition ExSpect calculates confidence intervals for all kinds of metrics (flow time, utilization, etc.). It is also possible to modify automatically-created simulation models of the workflow to support management games. A trial version of ExSpect can be downloaded from http://www.exspect.com.

### 5.5.5 BPR tools

In chapter 3 it was shown that there is a close relationship between Business Process Re-engineering (BPR) and workflow management. Therefore there are also links between tools to support BPR efforts and workflow management systems. Some of the tools supporting BPR efforts focus exclusively on simulation. ExSpect is an example of such a tool. Other tools focus on the modeling of business processes without any real support for analysis. Examples of tools that focus exclusively on modeling are Protos (Pallas Athena BV, Plasmolen, The Netherlands) and ARIS (IDS Scheer AG, Saarbrucken, Germany). Some tools offer both simulation and extensive modeling capabilities tailored towards business
Figure 5.19: Analyzing an erroneous workflow process developed using COSA.
ExSpect simulating a workflow process developed using COSA
processes, such as BusinessSpecs (IvyTeam, Zug, Switzerland), Income (Promatis AG, Karlsbad, Germany), and Meta Workflow Analyzer (Meta Software, Cambridge, MA., U.S.A.). To illustrate the functionality of these tools we briefly introduce Protos.

Protos Protos is a tool that can be used to model and document business processes. The tool is easy to use and is particularly useful for modeling workflow processes, that is, case-driven processes. Although Protos is not based on Petri nets it can support the diagramming technique used in this book. Protos supports the graphical modeling of processes, documents, applications, roles, groups, and teams. The analysis capabilities of Protos are limited: only very basic static dependencies can be analyzed (e.g., a role/route analysis comparable to the swim lanes in UML). Protos has excellent reporting facilities. It is possible to automatically generate RTF documents and HTML pages with hyperlinks. Protos supports an export facility to the simulation tool ExSpect. There also are interfaces with workflow management systems such as COSA (Ley GmbH), Corsa (BCT), and FLOWer (Pallas Athena). Figure 5.21 shows a screenshot of Protos. For more information we refer to http://www.pallas-athena.com.

5.5.6 Selecting a workflow management system
Selecting a workflow management system is not an easy matter. There are many aspects that need to be borne in mind. The selection process begins with the listing of the requirements that the system must meet. Based upon these, a shortlist is then compiled. When doing so, consideration is given to characteristics which are easy to check, such as the reliability of the supplier and whether the desired operating system and database management system are supported. The shortlist should preferably contain about five systems.

Each package on the shortlist is then subjected to closer scrutiny. One way to gain a good impression of a workflow management system quickly is to work through a sample process chosen in advance. Most suppliers are prepared to cooperate with a potential purchaser in doing this. It is very important that the sample process is representative of the relevant business processes. For example, one should ensure that all the
desired routing constructs are included. The sample process can be used to test both functional and performance requirements.

Figure 5.22 illustrates a possible sample process that, for the sake of convenience, we shall call P. Process P can be used to check functional requirements. All forms of routing are included, and a range of different triggers is used. The process is rather small for studying the performance of a workflow management system. However, if we produce a process in which P recurs four times as a subprocess, then we create something with far greater scope. By comparing the performance of the system when the four subprocesses run in parallel (linked by an AND-split and an AND-join) with that when there is selective routing between them (the four subprocesses are linked using an OR-split and an OR-join), one
Figure 5.22
Sample process for evaluating a workflow management system
can gain a good insight into the speed of the workflow engine. In both cases the full process consists of ninety tasks. This is sufficient for most applications.

Once the workflow management systems on the shortlist have been put on trial in this way, it usually becomes clear which package is the best choice.

5.6 Adaptive Workflow

5.6.1 Workflow management and CSCW
At the moment, there are more than two hundred workflow products commercially available, and many organizations are introducing workflow technology to support their business processes. It is widely recognized that workflow management systems should provide flexibility. However today's workflow management systems have problems dealing with changes. New technology, new laws, and new market requirements may lead to (structural) modifications of the workflow process definition at hand. In addition, ad hoc changes may be necessary because of exceptions. The inability to deal with various changes limits the application of today's workflow management systems.

Figure 5.23 shows the different fields of support for collaborative work. We distinguish between unstructured, information centric approaches (computer-supported, cooperative work or CSCW) and structured, process-centric ones (production workflow). Existing tools are typically in one of the two extremes of the spectrum: groupware products such as Lotus Notes and Exchange are typical CSCW tools, not providing much process support, whereas commercially available (production) WFMSs such as Staffware, COSA, and MQ Series are not able to cope with unstructuredness.

Linking production workflow management systems to groupware products does not really solve the problem, as the process logic then is still handled by the same inflexible workflow engine. To bridge the gap between CSCW and production workflow, several research groups are working on the problems associated with adaptive workflow. Adaptive workflow aims at providing process support like normal workflow systems do, but in such a way that the system is able to deal with certain changes. These changes may range from ad hoc changes such as changing
the order of two tasks for an individual case (often called exceptions) to the redesign of a workflow process as the result of a business process redesign (BPR) project.

Typical issues related to adaptive workflow are:

- **Correctness.** What kind of changes are allowed and is the resulting workflow process definition correct with respect to the criteria specified? We distinguish syntactic correctness (e.g., are there any unconnected nodes in the graph?) and semantic correctness (e.g., can existing cases in the system be finished in a proper way?).
- **Dynamic change.** What is done with running instances (cases) of a workflow of which the definition has been changed? The term dynamic change refers to the problems that occur when running cases have to migrate from one process definition to another.
- **Management information.** How to provide a manager with aggregated information about the actual state of the workflow processes?

Taking these issues into account, a classification of the types of changes is presented.

### 5.6.2 Classification of change

This section deals with the different kinds of change and their consequences. Some of the perspectives relevant for change are:
• **process perspective**, that is, tasks are added or deleted or their ordering is changed,
• **resource perspective**, that is, resources are classified in a different way or new classes are introduced,
• **control perspective**, that is, changing the way resources are allocated to processes and tasks,
• **task perspective**, that is, upgrading or downgrading tasks, and
• **system perspective**, that is, changes to the infrastructure or the configuration of the engines in the enactment service.

For workflow management systems, the process perspective is dominant. Therefore we focus on the process perspective when classifying the different types of workflow change.

First of all, we can classify change based on the scope or impact of the change. Using this criterion, two kinds of change are identified:

• **Individual (ad hoc) changes.** Ad hoc adaptation of the workflow process: a single case (or a limited set of cases) is affected. A good example is that of a hospital: if someone enters the hospital with a cardiac arrest, the doctor is not going to ask him for his ID, although the workflow process may prescribe this. Within the class of ad hoc changes it is possible to distinguish between *entry time changes* (changes that occur when a case is not yet in the system) and *on-the-fly changes* (while in the system, the process definition for a case changes).

• **Structural (evolutionary) changes.** Evolution of the workflow process: all new cases benefit from the adaptation. A structural change is typically the result of a BPR effort. An example of such a change is the change of a four-year curriculum at a university to a five-year one.

There are three different ways in which a workflow can be changed:

• the process definition is *extended* (e.g., by adding new tasks to cover process extensions),
• tasks are *replaced* by other tasks (e.g., a task is refined into a subprocess), and
• tasks in the process are *reordered* (e.g., two sequential tasks are put in parallel).

If a change occurs, it may affect running cases. Handling existing cases in the system when a process definition changes poses potential problems. Dealing with existing cases is only relevant in the case of a structural change because individual changes will always be (similar to) exceptions and as such will be dealt with by the one who initiated the change
explicitly. For structural changes there are three alternatives: (a) **restart**: running cases are rolled back and restarted at the beginning of the new process, (b) **proceed**: changes do not affect running cases by allowing for multiple versions of the process, and (c) **transfer**: a case is transferred to the new process. The term *dynamic change* is used to refer to the latter policy.

5.6.3 InConcert

Currently many researchers are working on problems related to adaptive workflow. Few commercial systems provide support for adaptive workflow. The problems related to dynamic change are difficult to tackle and not addressed by any of today's systems. Only for individual change there are some systems available. These systems are *ad hoc workflow* systems. In this section we describe one of these systems.

InConcert (TIBCO Software Inc.) is a workflow management system designed to develop flexible workflows. The tool has two unique features. First of all, the system supports "workflow design by discovery." This feature allows for the creation of templates based on the actual execution of workflow tasks for a given case. Second, InConcert supports a notion of class hierarchies that enables one InConcert object to inherit functionality of another InConcert object; in other words, the attributes of a parent workflow process definition can be inherited by child workflow process definitions.

Using the InConcert client software it is possible to bring into play the following tools:

1. **Process Designer.** The Process Designer is the tool used to design workflow process definitions. This tool can also be used to modify workflow process definitions on the fly.
2. **Task User Interface Designer.** The Task User Interface Designer is used to design the graphical interface presented to users when executing tasks.
3. **Work Group Manager.** The Work Group Manager is used to define new work groups and to monitor the workload of groups.
4. **Process Manager.** The Process Manager is used to start and manage cases (workflow instances).
5. **Document Organizer.** The Document Organizer is used to organize and create InConcert documents.
6. **Task Organizer.** The Task Organizer is used to display and execute work items.

Figure 5.24 shows the Process Designer of InConcert. The modeling language used by InConcert corresponds to a subclass of Petri nets: Acyclic Marked Graphs (AMG). This is the class of Petri nets without any cycles, and each place can have neither multiple input transitions nor multiple output transitions. InConcert does not provide any explicit OR-splits and OR-joins. Every task is considered to be an AND-split and an AND-join. To enable conditional routing each task has a Boolean condition associated to it: the so-called *perform condition.* The perform condition can be used to skip tasks. The workflow design shown in figure 5.24 shows the process of handling insurance claims. The task *pay* has a perform condition indicating that it should only be executed if the outcome of task *decide* was positive. The check tasks in figure 5.24 also have a perform condition: either the two parallel checks (top) or the three sequential checks (bottom) are executed.

The fact that InConcert does not allow for OR-splits, OR-joins, and iteration simplifies the modeling process. Workflow designers cannot make workflow models that deadlock or never end: The workflow process definition is guaranteed to be sound (cf. chapter 4). This makes InConcert a system where end-users can design or modify process definitions. Unlike production workflow management systems, InConcert associates a unique process definition to each individual case (i.e., workflow instance). There are several ways to create a new workflow instance:

1. Instantiate an existing workflow process definition: a copy is made of the process definition, and the first task is enabled without changing the workflow.
2. Instantiate a customized version of an existing workflow process definition: a copy is made of the process definition and is changed to allow for ad hoc routing.
3. Instantiate an ad hoc workflow process by specifying a sequence of tasks and users.
4. Instantiate a so-called "free routing process," that is, an empty ad hoc workflow process. There is no explicit workflow process definition: the workflow is created on the fly.

Instantiating an existing workflow process definition corresponds to the way cases are handled in traditional production workflow systems. The
Figure 5.24
The Process Definition tool of InConcert
only difference is that the case does not refer to a common workflow process definition but to a private copy of the definition. By creating a copy and the possibility to change that copy, either at creation time or on the fly, the workflow process definition serves as a template. Instead of creating a copy of such a template, it is also possible to create an ad hoc process from scratch. The fact that each workflow instance has its own workflow process definition allows for on-the-fly changes. In principle, it is possible to modify the routing of a case at any point in time. This way ad hoc changes are supported completely. In addition, InConcert supports "workflow design by discovery." The routing of any completed workflow instance can be used to create a new template. This way actual workflow executions can be used to create workflow process definitions. Figure 5.25 shows a screenshot of InConcert while changing the process definition of a running instance.

InConcert also supports a class concept. There are three types of classes: process classes, task classes, and document classes. These classes are grouped into a class hierarchy and a child class inherits the attributes of its parent class. The class Job is the parent class of any process definition. By defining a child class Activity_based_costing_processes, all standard attributes are inherited and new costs attributes can be added. Any process definition of this new class is equipped with these new attributes. Similarly it is possible to define task and document classes. The class concept encourages reuse and a uniform way of realizing workflow support.

5.7 Workflow Management Trends

At present there are many suppliers of workflow management systems. The products they market are still developing at a rapid pace. It is a trait seen with all generic software: the manufacturers are, as it were, in a race. Each one tries to incorporate its competitors' successful functions into a new version of its own product as soon as possible, as well as devise some new features of its own as unique selling points. Thanks to these developments, we can see the packages converging with one another—although there are still differences. It is clear that the functionality desired by the WFMC is still far from being achieved. Nor is there enough practical experience as yet for us to know precisely what
Changing the process definition of an instance on the fly
functionality workflow management systems will eventually encompass. It therefore is interesting to summarize their future potential.

As we shall see, workflow management systems have many application possibilities. But this also represents a threat, since the manufacturers of other generic software components—such as database management systems and logistical/ERP packages—will also incorporate workflow management functionality into their own products, eliminating the justification for the existence of separate workflow management systems.

We shall examine the future prospects for workflow management systems, in terms of opportunities and threats, in terms of seven areas of functionality:

1. Modeling;
2. Analysis;
3. Planning;
4. Transaction management;
5. Interoperability;
6. Internet/Intranet; and
7. Logistical management.

Because specific software for each of the above is also available, we shall consider threats alongside opportunities (i.e., application possibilities).

5.7.1 Modeling

One of the most important functions of a workflow management system is the modeling of workflows. This ability means that such a system can be regarded as a repository for metabusiness data: an organization's structural information, such as its processes and organizational diagram. Such tools have been given the name orgware (from "organization-ware"). However, there are specific repositories in which much more of an organization's data can be recorded: for example, all kinds of performance indicators of business processes, a corporate data model of the organization (a "data dictionary" of all the databases which it uses), and a roadmap of its information systems.

The advantage of such repositories is that they offer good query opportunities through which all the connections relevant to the management of the organization can be analyzed. They are often developed using a database management system and/or an OLAP tool (on-line analytical
processing). One essential difference between these is that OLAP tools enable hierarchical structures to be searched through recursively (known as "downdrilling"), which is not possible in SQL (the query language used in relational database management systems). It therefore is obvious that workflow management systems will acquire more repository functions in the future, or improved interfacing with such tools.

Another important aspect is the expressive power of the modeling function in the current generation of workflow management systems. Many of the existing systems do not have a good process model. This means that certain common constructions in business processes are not handled well. This problem will certainly be solved, and one can expect that eventually all workflow management systems will model their processes in a way concomitant with the Petri-net theory.

One final aspect of modeling is that today's workflow management systems are mainly suited to standard processes. In other words, the process definition tool describes a number of business processes, by which many cases are performed. Because the number of cases is in general relatively large compared with the number of processes, we refer to this as production workflow. In the future, however, we should also expect systems which offer functionality for so-called one-of-a-kind processes (ad hoc workflow), with a separate process defined for each case.

An additional complication is that processes may change while a case is being processed. One encounters examples of this in the transport industry (when decisions to change route are made on the road) and in healthcare (when the appropriate treatment can only be decided after the diagnosis phase). In present-day workflow management systems, this can be partially overcome by defining a process with very generic tasks—but this is only really shifting the problem. The use of generic tasks results in much of the management having to be done within the applications. Solving the problem will mean further integrating the process definition functionality with the workflow engine.

5.7.2 Analysis

New business processes are analyzed in order to establish whether they will perform well in both the quantitative (completion times, resource utilization, and so on) and qualitative sense (are they correct, i.e., sound, and workable for the people in the organization?). When existing pro-
cesses are improved, analysis of the modified processes is also desirable before the changes are put into effect. To perform analyses, we can use simulation and several formal verification techniques. Further expansion of these abilities is an obvious future development. For simulation, this means that it will be made easier to use historical data from the workflow management system to test modified business processes, and more opportunities for "games" will appear. In other words, people who play a part in the processes seek out weaknesses in the workflow management system using a business simulation game. This function can also be used to train new staff. Several existing workflow management systems already offer some game facilities, but there is much scope for improvement—for example, supporting rollback capabilities.

There are many simulation tools on the market, and it is not unthinkable that these will develop in the direction of workflow management systems. After all, it is not such a great leap from simulating workflows to coordinating real ones. It therefore is possible that some simulation tools may evolve into workflow engines. As well as simulation, there are also the formal analysis methods, which still leave a lot to be desired. Those available have mainly been developed for Petri nets and are not geared to specific business-process structures. It is likely that several correctness tests, like the ones offered by Woflan, will be incorporated into the process definition tools in the future. These will "rap the designer's knuckles" if he makes an error, without him having to understand the theory underlying the tests.

5.7.3 Planning

The current generation of workflow management systems sometimes offers only a limited ability to allocate resources to tasks and to decide the order in which tasks using the same resources should be carried out. (This type of planning is known as scheduling.) Existing systems pay virtually no attention to the timetabling problems that occur when organizing human resources. And owing to increasing labor flexibility and organizations' lengthening hours of business, this problem is becoming more and more significant. Functionality is required which is at present not sufficiently supported by workflow management systems.

Better planning support may be offered by the application of modern operations research and artificial intelligence methods in the preparation
of rosters and schedules. Such methods as *simulated annealing*, *taboo search*, and *constraint satisfaction* have proven themselves in practice in recent years. Alongside these *operational* planning problems there are also *tactical* ones that pertain to decisions about how much of the *capacity* of particular resources (not just human ones) will be required during the period being planned for. Although a workflow management system does in fact contain all the relevant information needed to solve such problems, none yet actually offers the facility to do so. Also at issue is whether the producers of these systems should develop such functionality themselves, or whether it would be better for them to try to integrate propriety planning software into their programs.

### 5.7.4 Transaction management

Thus far most workflow management systems have confined themselves to work processes within a single organization. In doing so, they assume that the (human) resources are employed exclusively by that organization and can be allocated at will by the resource management (the boss or the workflow management system). Consequently it is assumed that all the human resources have the same client software and that all information exchange with them occurs in a uniform way. If we wish to apply workflow management systems to coordinate business processes in virtual companies or network organizations, then various problems arise that cannot be tackled by the current systems. Note that workflow management systems are very relevant for supporting e-business transaction processing. However they need additional functionality to support inter-organizational processes.

As described in chapter 1 using an example from the transport industry, finding a suitable resource will require a *communications process*. In doing so, a transaction tree is passed through until an actor is found who is willing to perform as a resource. An additional complication is that we can no longer assume that all the resources are able to interpret the same information. Messaging standards and conversion software like those commonly used in the EDI world therefore will become vital in inter-company workflow. XML offers a very promising standard for this. The communications process between the parties involved will not only cover the time within which the task can be completed, but also the amount of money associated with it. So workflow management systems will also
have to provide functionality for the financial settlement of the work performed by resources.

One interesting complication of workflow management within network organizations is that the term "task" changes. It is not an atomic piece of work for everybody. What is a task for the principal is a process definition for the contractor. This is why it is so good that we use hierarchical Petri nets, because they can model such situations with ease. If the transaction trees (see chapter 1) for finding suitable actors to perform the case become very high, and each actor will only offer an upward commitment (a confirmation of order to its contractor) once it has obtained such a commitment from its subcontractors, then acceptance of an order at the highest level can become an extremely time-consuming business. This forms a "natural threshold" for the effectiveness of network organizations. In some situations, they will be practicable only if the communications process can be made largely automatic. As well as messaging standards, comprehensive agreements between the actors are also required to achieve this. Moreover, the additional functions for workflow management systems in network organizations also will bear fruit for hierarchical ones. After all, they provide an opportunity for controlled decentralization and so empower employees.

5.7.5 Interoperability

One of the interesting properties of a workflow management system is that human resources and computer applications are treated in a uniform way. The system organizes all the work that needs to be carried out on a case. In other words, it deals with the scheduling of resources and ensures that they have the correct information when they begin performing the task. In short, the workflow management system provides the logistical management of the work, and so closely resembles a computer operating system. After all, the operating system also performs tasks for the various user transactions and batch jobs. The difference is that a workflow management system also controls the work of human resources that are outside the computer system.

A workflow management system thus can be regarded as a kind of operating system for an organization. In theory, it could also be used to link various computer applications, since the order of tasks is described
by the work process as some kind of flow chart. Such a system therefore
could perform the control flow of a large information system, with the
application programs carrying out its data transformations. However,
although possible in principle, the current generation of workflow man-
agement systems is not yet suited to this type of usage. First, the existing
standard application programming interfaces (APIs) are too limited. Sec-
ond, the workflow management system would have to be able to func-
tion as a kind of software bus between various applications—a role for
which its performance is still quite inadequate. It also would have to be
possible to monitor protocols between communicating applications and
to support data conversion between them. Moreover, there is often no
functionality for rolling back transactions and coping with hardware
failures. If these restrictions could be overcome, a workflow management
system would become an ideal tool for bridging interoperability problems.

5.7.6 Internet/Intranet
A limited number of workflow management systems allow the use of a
web browser such as Netscape Navigator or Microsoft Internet Explorer
as a workflow client application (Interface 2). In such cases, a
system-specific worklist handler is not used; instead the browser acts as
the worklist handler. This makes it possible for us to access the
workflow system through the Internet, also known as the World
Wide Web (WWW). This has a number of significant advantages. First,
one is no longer confined to a particular workplace. If the workflow
management system is linked to the WWW, then in principle it is
possible to perform work anywhere. Even from Australia, for example,
there is no problem accessing a workflow system in Europe.

Another important advantage is the fact that one can employ widely
accepted standards such as HTTP (HyperText Transfer Protocol),
HTML (HyperText Markup Language), XML (eXtendible Markup
Language), and CGI (Common Gateway Interface). As a result, there is
no dependence upon exchange protocols specifically developed for a par-
ticular workflow management system. The use of XML/HTML pages is
sufficient. The combination of workflow and the World Wide Web opens
up new application opportunities: e-business. Many services offered on
the Web can be supported by a workflow management system. Consider
for example the processing of orders, complaints, applications, and so on. Interestingly, these applications blur the distinction between customer and employee: both access the workflow system in the same way. However there are also some problems associated with the use of the World Wide Web as a workflow client application. First, its speed may leave much to be desired; it often takes some time before a task can be opened or closed. Nor is the security perfect. Confidential information is difficult to protect. These problems can be solved to a large extent by using an Intranet. This has the same structure as the World Wide Web, but is limited in extent. Consequently a company can "shield" its network from the outside world and speeds are not limited by the "traffic jams" on the World Wide Web. Nevertheless it remains possible to use the standards and products mentioned above.

One problem that cannot be solved by an Intranet is the ponderous use of applications. Interactive applications such as word processors can only be started up through additional facilities, and data-intensive applications result in high loading of the network. New development environments (such as Java and CORBA) can only partially solve these problems. It therefore remains unclear what perspectives the World Wide Web can offer the future generation of workflow management systems.

5.7.7 Logistical management

One of the most successful categories of generic applications is that of logistical management systems, also known as ERP Systems (enterprise resource planning systems). Some of these packages have evolved from financial software and developed further through the extension of the stock-administration functionality. They enable the support of a large number of business functions in production (e.g., the automotive industry), distribution, transportation, discrete manufacturing, banking, insurance, and government. One of their most important functions is the calculation of materials requirements, based upon the planned lead-time of a product. Conversely the materials requirements are used to generate, a detailed schedule. The basis for this is a products component list, also known as the "bill of material" (BOM). If a product must be ready on a certain date and it is known how long it takes to put together its largest subassemblies and finish the product (for example, paint), then one can
calculate when the subassemblies must be ready. If they are also made in
house, a similar schedule can be drawn up for the subassemblies. If they
are purchased externally, a delivery deadline can be set.

The current generation of logistical management software does not use
the term "business process" as generically and flexibly as today's work-
flow management systems do. Naturally their vendors follow develop-
ments in workflow management systems closely and are likely to
incorporate some of the workflow functions in new generations of their
products. Whether, given the structure of their products (legacy), they are
able to do this effectively is difficult to foresee. Certainly such products
have many other very interesting functions—particularly for production
companies—and could probably compensate effectively for rather weak
workflow support.

This threat again has an opportunity as its "flip side": it is quite pos-
sible to incorporate a number of functions from logistics packages into
workflow management systems. The bill of material is of particular
interest. Workflow management systems are always based upon a pro-
cess made up of a number of tasks. The precise content of these tasks is
entirely ignored, as is the information required to carry them out.
Drawing up a bill of material for each type of case showing what infor-
mation is required to complete it would in theory enable one to *deduce*
what the tasks are. We can illustrate the use of such a list using the
insurance claim example from chapter 1. The case can be closed when
the level of payment is known and when the policyholder has agreed to a
settlement (which may be zero). The amount therefore is required, and
for this the value of the claim must be established, as well as whether it
meets the policy conditions. (And so we can go on.) In this way, one can
deduce the process from the information needs and have the format of
the data required for each task immediately at hand. By beginning with a
bill of material, the process designer can start her work at a higher level.
This list can also be useful for the workflow engine, by enabling it to
gather the information it requires in advance and to submit this to the
resource at the appropriate moment.

We now have seen seven groups of functions that will be of importance
to the workflow management systems of the future. Some already are
being incorporated into the latest generation of systems. It is unlikely,
though, that manufacturers will incorporate all this functionality. This would not be sensible, because they would never be able to remain up to date in every one of these fields. A better solution is for the architecture of their systems to be left sufficiently open so that it is easy to integrate other manufacturers' software packages—with specific functions from the range described—into them. But for this a great deal of standardization is required.

EXERCISES

Exercise 5.1
Describe the reference model of the WfMC; that is, provide a graphical model of the components and interfaces. Describe each component in detail. Also discuss the functionality of each of the five interfaces.

Exercise 5.2
Answer the following short questions:
(a) What are the ACID properties?
(b) Which interface typically causes technical problems?
(c) What are the four roles of people involved in the design and deployment of a workflow management system?
(d) Name some examples of workflow interoperability standards focusing on run-time aspects.
(e) Characterize the following workflow management systems: Staffware, COSA, and ActionWorkflow.
(f) What is the functionality of analysis tools such as Woflan and ExSpect?
(g) Name some BPR tools.

Exercise 5.3
Model the process shown in figure 5.26 using the modeling languages supported by Staffware and COSA.

Exercise 5.4
Model the traveling agency described in chapter 2 using the modeling languages supported by Staffware and COSA.