2

Background Knowledge

This chapter describes the general background that sets the context for the work that is described later. It firstly introduces an overview of various *Enterprise Modelling (EM)* methods which are currently available. It then discusses *Business Modelling* in the *Business System Development Method (BSDM)*, the method for which a formalisation is proposed in this book. An introduction to *Business Process Modelling (BPM)* follows, since some of the extensions to BSDM used in this work are based on techniques developed in the area of BPM. There is also a brief discussion of software systems and other related work.

The purpose of this chapter is not to cover all the background needed in great detail, but to set the scene for the rest of the book. More details of BSDM are introduced in subsequent chapters together with the corresponding formalisation work. We also defer a brief introduction to *Case-Based Reasoning (CBR)* techniques to Chapter 8, where the use of CBR in KBST-BM is explained.

2.1 Why Enterprise Modelling?

The global economy and market in which a business operates and competes has changed so dramatically in the last decade that traditional business management and operational methods are no longer sufficient to manage today's business. Three main driving forces are behind these changes: rapid advances of modern computing technology, intensified competition of the world market, and changing demands from consumers [85].

The advances of modern (computing) technologies have continuously provided companies with new ways to do their business: both internal and external to the organisation. Internally they provide more standardised, efficient and direct control over the working processes which are supported with organised information that is easily sharable among relevant personnel. Externally they provide a revolutionary medium to interact with customers and other businesses. It is no longer necessary to face the customer in person or to provide a shop floor. Communication with customers and other businesses, such as order-

ing goods and delivering services, can often be done electronically (e.g. through the Internet).

Modern computing technology and the fact that the world market has become more accessible and exploited by businesses have made the world smaller and the competition for customers more acute. It is commonplace that similar kinds of services and goods are produced by companies all over the world. The boundaries set by countries or geographical distance have become less important. Customers can now more easily shop around companies all over the world to get the best product at the best price. To gain a competitive edge, the modern enterprise is a virtual entity which consists of many sub-organisations spread across many different geographical areas each with special functionalities and business advantages. Customer demands have also changed. Customers today are more informed and aware of their power. They are no longer satisfied with mass produced indifferent goods or passive services. Instead they demand more sophisticated and individualised products and better and quicker service. This puts pressure on companies to offer high quality, diversified customer-tailored goods and services, and at the same time to offer them at a reasonable price and delivered within a relatively short time.

All of these demands require a radical change in how a business operates. It not only needs to acquire stronger financial backing to be able to compete on the global market, it also needs to create wider and intense direct contacts with (potential) customers. Some companies achieve this through the Internet, some by gaining more business allies, others by becoming larger companies through merging. More importantly, to cope with these changes, a business needs to introduce and practise brand new sets of management and organisational methods. In fact, these changes have forced many of today's businesses into fundamentally **rethinking and redesigning** their strategies and operations. Instead of adapting various ad-hoc solutions on a trial and error basis, companies seek methods which help them to analyse their businesses as a whole systematically and effectively, which in turn help them improve organisational performance. To address this problem, *Enterprise Modelling* methods have been deployed.

2.2 Enterprise Modelling Methods

A variety of enterprise modelling methods have emerged during the last decade. They provide a structural framework to help an enterprise capture the enterprise-wide knowledge which forms the basis for the targeted analysis and helps the re-shaping and re-designing of a business. A key goal of applying these methods is to seek ways to improve an organisation's effectiveness, efficiency and profitability.

Most enterprise modelling methods are influenced by more than one discipline and often overlap with each other in some aspects. It is therefore difficult to gives an absolute classification for them. Nevertheless, this information is useful in understanding the different EM methods. We therefore try to cate-

gorise them in three groups depending on their origin, the application domains that use them actively and the way that they are used in a broad sense. The three types that have been identified are:

- business process modelling,
- business system modelling, and
- organisation context modelling methods.

Business Process Modelling (BPM) methods were initially inspired by process modelling techniques which provide precise formats to capture processes that are practised in a manufacturing environment. By using these techniques, informally practised processes can be made more concrete and formal analysis of processes can be carried out. More importantly, actions and effects of these processes can be demonstrated using simulation techniques. The performance of each process can therefore be predicted and used to choose between competing processes [85].

These techniques have been adapted and extended by *business process modelling* methods to capture and standardise processes practised in a non-manufacturing environment. This has enabled the analysis and re-design of processes in the service sector leading sometimes to radical performance improvements. Representative business process modelling methods are described in the *Handbook of Organisational Processes* [70], *Workflow Reference Model* [49], *Process Interchange Format (PIF)* [62], *Process Specification Language (PSL)* [98],[1] *Integration DEFinition Language (IDEF3)* [72], *Integration DEFinition Language (IDEF0)* [80], *UML's Activity Diagram* (extension) [94], *Event-driven Process Chains (EPC)* [89] and *Petri-Nets* [87].

In addition, based on fundamental (business) process modelling concepts, new process languages are being developed to promote understanding and interoperability of process semantics over the Internet (and the *Semantic Web*).[2] These languages are characterised by their chosen representations that are based on XML, RDF or OWL.[3] They may also provide constructs to assist communication between processes over the Internet. This is a field that is relatively young and languages in flux. Examples of such languages are *Web Services Business Process Execution Language (BPEL)* [4], *ebXML Business Process Specification Schema* [64], *XML Process Definition Language (XPDL)* [117], *Business Process Modelling Language (BPML)* [5], and *OWL-based Web Service Ontology (OWL-S)* [71].

Because some of the process modelling techniques used by the above methods have a strong influence on the work of this book, Section 2.4 will describe

[1] PSL and PIF provide a common platform for communications between process modelling languages.

[2] Semantic Web is a (conceptual) layer on the Internet. It consists of data and applications where meaning (or semantics) is encoded to support knowledge sharing.

[3] XML, RDF and OWL are representational languages that have been designed to describe the semantics of data and to support machine processing (vs. human understanding) over the Internet.

them in more detail. As process languages that have been developed for applications over the Internet are often a use of the process modelling concepts that are already included in the fundamental methods, they will not be discussed further in this book. Readers who are interested in more details are therefore referred to the above citations for more information.

The creation of **Business System Modelling (BSM)** methods was inspired by the software engineering community where discrepancies were recognised between the vision of software engineers for the software system to be built and the true need of a business for its procured software system. The motivation for employing *BSM* methods is often to provide a clearer picture and directions for building a better IT system [53].

BSM methods provide the means to describe a business and capture its operations from a business point of view, not confined by technical, specifically Information Technology (IT), considerations. This means that for each business model there are potentially many different ways to implement a software system. Examples of BSM methods and techniques are: *BSDM*'s business modelling method [53], *ORDIT* [31], *Role Activity Diagram (RAD)* [82], *Meta-Model* by Scacchi et al. [75], *Swim-lane Diagram* by Rummler et al. [95], the *Business Modelling* approach using (extended) *UML* notation by Eriksson et al. [34] and by Rational [86], and the reengineering method developed by Jacobson [55].

Organisation Context Modelling (OCM) methods capture and tackle the wider organisational issues within a business. This includes methods which capture the functional, structural and/or cultural aspects of an organisation. It also includes methods which capture the decision making processes as well as the vocabularies and terms that are used in the business context. To promote effective organisational knowledge management and utilisation, Macintosh et al. [67] and Schreiber et al. [99] provide a framework to identify, obtain and maintain the required knowledge and skills for an organisation and the means to make use of them to achieve organisational objectives. Yu, Mylopoulos and Lesperance [118] provide graphical notations to capture business strategies and their rationale in the *Strategic Dependency* and *Strategic Rationale Models* which exploit links between business strategies and the actual operations. To promote better communication via a common language within and between organisations, ontologies have also been developed for businesses. Representative examples of work in this area are the *Enterprise Ontology* developed by Uschold, King, Moralee and Zorgios [113] [114] and *Tove* by Fox, Gruninger et al. [38]. Other OCM methods are *Activity-Based Costing (ABC)* [30], *Simulation Modelling* [100] and *Total Quality Management (TQM)* [29].

In parallel to the development of the above methods, techniques in *Business Process Reengineering (BPR)* have become a popular management tool for rapid enterprise re-structuring and re-design. Example literature includes Hammer et al. [45] and DOD [112]. Instead of using modelling methods, they provide a collection of generic business management principles coupling with software engineering methods. When deploying such BPR techniques, OCM methods, such as ABC and TQM, can also be used as a part of a BPR initiative – although they are often used in their own right. All of the above methods

provide the means to record and analyse some aspects of a business environment and therefore all of them can be used to support BPR initiatives. This book focuses on one EM method, the *Business Modelling* method in BSDM, which is described below.

2.3 Introducing the Business System Development Method (BSDM)

The Business System Development Method (BSDM) is an enterprise modelling method which was introduced by IBM [53]. It provides a modelling framework to capture and analyse a business operation and requirements which helps the understanding of the complex business environment as well as providing a basis for strategic analysis and re-structuring of the organisation. It also provides a specification for the design of a software system from an early stage from the business point of view and independent of any information technology considerations which makes the developed software system more "business-need-oriented" rather than "technology-oriented". The ultimate goal for applying BSDM is to improve an organisation's performance.

BSDM consists of four activities: *Map, Need, Shape* and *Run*. BSDM firstly describes business environments, its policies, components and constraints and represents them in a *Business Model* during the *Map* activity. Given this business model, BSDM then provides the means to identify and specify requirements for a business system during the *Need* activity.[4] Based on these requirements, BSDM then allows the user to choose any suitable (software engineering) methods to design and implement the business system during the *Shape* activity.[5] The actual deployment of the system takes place during the *Run* activity. Since the most distinctive and important activity of BSDM is the *Map* or *Business Modelling* activity, this book focuses on that activity.

2.3.1 Business Modelling

The main components of a BSDM *Business Model* are *Entity Model*, *Process Model*, *Life Cycle Diagrams*, and their supplemental textual descriptions. At the beginning, business managers together with a BSDM facilitator create an *Entity Model* which captures the concepts (abstract and concrete things) in the business as *entities* and the relationships between them as *dependencies*. The Entity Model is then extended with information about the *processes* which manage these entities to form a *Process Model*. A BSDM process describes the context of a business process, the circumstances which trigger such a process

[4] A suitable *business system* may not necessarily involve a computing system; it can be a computing system supported by a manual process, or it can be a purely manual system depending on the business need.

[5] A suitable method for designing and implementing a business system may be a software engineering method suitable for the organisation.

and the effects of its actions. In parallel to the development of entity and process models, *Life Cycle Diagrams* are built. They describe information about an entity's life statuses and how different processes manipulate these entities to enable transitions between these life statuses. They also indicate the subtle relationships between processes and the operations used to carry out a particular task.

BSDM provides step-by-step procedures for building business models with supporting recommendations, guidelines and example models. A business model is normally built during BSDM workshops over a few months. Conventionally, the model is initially paper-based. The graphical information is later recorded in a graphical tool and the textual information in a text-editing tool.[6] The quality of the built model relies entirely on the knowledge and experiences of the participants in the project. In Chapter 3, the kinds of automatic support which can be provided for such informal modelling methods will be proposed. The potential benefits of these kinds of automatic support will also be discussed.

2.3.2 BSDM Compared with SE

BSDM can be used in several different contexts, e.g. as a business analysis tool, a management tool, or as a support method before a software engineering method is carried out. To show the relation between BSDM and traditional *Software Engineering (SE)* methods, a comparison is given below.

The part which distinguishes BSDM from a conventional software engineering method is the *Map* or *business modelling* activity which captures and specifies business requirements that fills the gap between conventional software engineering (SE) methods and business modelling needs during the requirements analysis phase. Figure 2.1 shows how BSDM is mapped onto conventional software engineering methods, adapted from the comparison given by Spurr et al. [103].

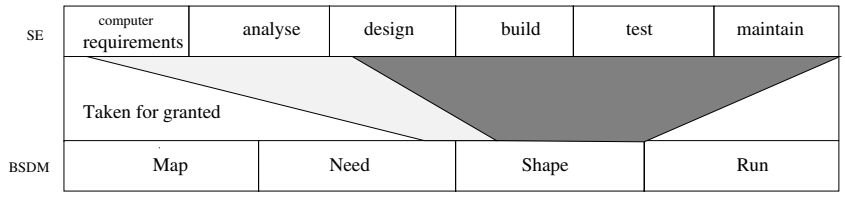

Fig. 2.1. A comparison of BSDM with conventional SE methods

The most significant difference is the lack of *business modelling* activities in traditional SE that is the mapping area labelled with "Taken for granted". This missing part represents the activities of identification and analysis done in *Map*; as well as the activities of evaluating requirements for a business need

[6] The graphical tool was not built specifically for BSDM and does not properly support its notation and use.

in *Need*. Since part of the BSDM *Need* activity is based on the earlier *Map* activity, much of it has no counterpart in SE.

The SE *computer requirements, analyse*, and the early stage of *design*, which include the capturing and analysis of user requirements[7] and the logical and architectural design of a software system, corresponds to the later stage of *Need* and the early stage of *Shape* activities in BSDM. Since the later part of *Need* includes user requirements capturing and analysis (for technical and operational issues), the scoping of a business system as well as the determination of main system functionalities, it corresponds to *computer requirements, analyse* and the early stage of SE *design* activities. The early activities of *Shape* include the logical and architectural design of a business system, and therefore are similar to the early activities of *design* in SE.

The rest of the SE phases, the late activities of *design* and the *build, test* and *maintain* of a software system, are mapped onto the BSDM *Shape* activity. The SE *maintenance* phase correspond to a revisit of *Shape* after the deployment of the system, and sometimes even a revisit of *Need* when necessary. Since *Run* indicates the actual use of a business system, it is not considered an SE activity, therefore it is not mapped onto any phases in the SE method.

The building of a software system is an iterative cycle that is sometimes described in a *Plan-Build-Test-Refine* spiral model. The business modelling activities can also benefit from the same principle. Chapters 3 and 9 illustrate how our automatic support can help this iterative modelling process.

2.4 Introducing Business Process Models

Although *BSDM's Business Modelling Method* is not a "process-oriented" modelling language, nor is it directly influenced by *Business Process Modelling (BPM)* methods, techniques used in BPM can be adapted and used to extend BSDM notations and can amplify and diversify the use of BSDM models. A brief background description of BPM is first given below.

Since the 1960s and 1970s, process modelling has been applied in the manufacturing sector [85]. Motor companies, such as Ford, and aerospace companies, such as McDonnell Douglas Corporation, have used process models to capture the processes of designing and manufacturing products. These process models were also simulated to allow predication and evaluation of trade-offs of current design, and used to guide the construction and selection of alternative designs.

The early acceptance of process models in manufacturing sectors was mainly due to the need for frequent change of products which requires frequent and rapid generation of production processes. It is also due to the fact that working procedures in a manufacturing environment are comparatively clearly defined and sometimes formalised. These useful characteristics initially were not obvious or were believed to be non-existent in the service sector: their procedures being more informal and open to interpretations that differed depending on the person who implemented the tasks.

[7] With regard to aspects of IT and the actual working procedure considerations.

This situation changed in the early 1990s when a great majority of informal business processes were found to be similar and repetitive, so they too can be captured, analysed and improved using modelling techniques similar to those of process modelling (Harrington et al. [46], Malone et al. [70]). This discovery encouraged the creation and use of process modelling methods in a more general business environment rather than in a pure manufacturing context.

Enterprise modelling methods which are evidently influenced by this are classified as *Business Process Modelling* methods in this book, as mentioned earlier in Section 2.2, which includes methods such as the *Handbook of Organisational Processes, PSL, PIF* and *IDEF3*. All of these methods treat the processes practised by an organisation as the central focus in their modelling activities. To help understand what a process is, Chris Menzel's definition is given below:

> An objective real world event, described totally as a sequence of events (activities, sub-processes) occurring over time containing certain objects having certain properties standing in certain relations. [98]

This is further elaborated by Jeffery Herrmann:

> A process can be decomposed into other processes. A process begins and ends at points in time. One can view a process from different perspectives that include different things. Objectives or drivers may be part of one perspective but not another: if included, they could be seen as instructions. [98]

The descriptions of a process given above are applicable for processes in many process models including the one described in this book. A process is, therefore, an event which may include many activities where each activity may also be itself a process that is decomposable – this is the *decomposability* property. A process often lasts for a period of time during which it may involve the manipulation of various objects as well as actors who enact or interact with it at some point of time. It is, therefore, necessary to identify and represent those temporal relations between those objects and actors and the corresponding processes in a process model. These characteristics and their representations will be discussed later in the section on PIF and in our formal work in Chapter 7.

In summary, in a (hierarchical) process model,[8] processes described at a higher level can be divided into *sub-activities*. These sub-activities carry out collaborative and complimentary activities so that together they accomplish the higher level task. These sub-activities also provide more implementation details towards the task. In addition, sub-activities may again (recursively) be divided into even smaller tasks and described in further detail. This is the concept of **process decomposition** [72] [70].

In some methods, decomposed (or sub-) processes may include *alternative processes*. Those processes are used to describe a process from different view

[8] There are some (and not many) process modelling methods can only describe flat level processes.

points, e.g. from a neutral observation view point, or from a particular actor's view point [72]. This concept has been extended in this book and *FBPML* [25] where **alternative processes** are competing sub-processes that achieve the same purpose as a common generic task, but may involve different working procedures, objects and/or actors. For instance, for a business to receive a payment from its customer, it may receive it using different methods, e.g. over a counter, the phone or via the Internet. Each method may involve different working procedures and objects and may also have different actors. So long as a business receives a payment, regardless of the method used, the more generic process "Receive payment from customer" is accomplished.

In parallel, in the *Process Handbook project* [70], once processes are identified for an organisation, they are classified and represented in a *class hierarchy* (of specialisation). Processes represented at a higher level of abstraction in the class hierarchy may be *specialised* into *sub-typed* processes. For instance, the more generic process of "Sell product" may be specialised into "Sell by mail order" and "Sell in retail store" where more details are added. As in object-oriented programming, processes described at the higher level of the hierarchy describe more generic tasks, and they often possess characteristics and properties that are sharable by processes described at the lower level of the class hierarchy. These common properties can be passed to or *inherited* by the more specialised processes described at the lower level of the hierarchy. This is **process specialization**.

The benefits of *process specialisation* and *decomposition* are essentially in four areas. Firstly, these reduce the work of developing a new process. By identifying an appropriate position in a class hierarchy for a new process, fundamental features of that process can be automatically inherited from existing processes which are at a higher level of the hierarchy. Secondly, they can decrease the work for maintenance: any error only needs to be corrected once at the highest level and all of the more specialised processes are corrected. Thirdly, since all similar processes are grouped together, it is easier to evaluate the trade-offs and select between them. Lastly, by providing a taxonomic structure, process allocation, searching, combining and creating of new processes can be done more systematically and efficiently. This combined use of *process decomposition* and *specialisation* was first identified in a process modelling framework [109][9] and used in the *Process Handbook Project* and was accepted and used in many later developed modelling methods.

In this book, the concept of *process specialisation* has been used to show how BSDM's business processes can be classified and reused by incorporating them as a part of our *Inheritance Class Hierarchy* and will be introduced in more detail in Chapter 5. The concept of *process decomposition* has also been adapted and used in our devised *Procedural Model* which will be described in Chapter 7. By deploying these concepts, our formal architecture is able to enjoy many of the above benefits, such as inheriting properties from a more general process to a more specialised one, ease of maintenance, comparison

[9] Through its use in hierarchical planning techniques in AI dates from the mid 1970.

and manipulation of processes. Temporal and other constraints which limit the execution of a task are also included in a process model to describe and prescribe the implementation of the actual working practice. This information may be encoded using dependencies and junctions between processes as well as attributes of processes that will be demonstrated in Chapter 7.

Although it can be used to capture (business) processes, the main purpose of the *Process Interchange Format (PIF)* is to provide a common language that enables different process models to communicate and exchange information through them. PIF identifies a set of concepts that are fundamental to process modelling and is commonly used in many different process models. Based on these concepts, PIF gives precise descriptions for each concept and defines the relationships between them. Formats based on a *frame structure* are also provided by PIF to capture and store information of these concepts – this set of fundamental concepts for process modelling is called *PIF's core*. Specialised processes that are captured in other process modelling languages which cannot be described using only PIF's core may be represented using an extension format that is described in *PIF's Partially Shared View*.

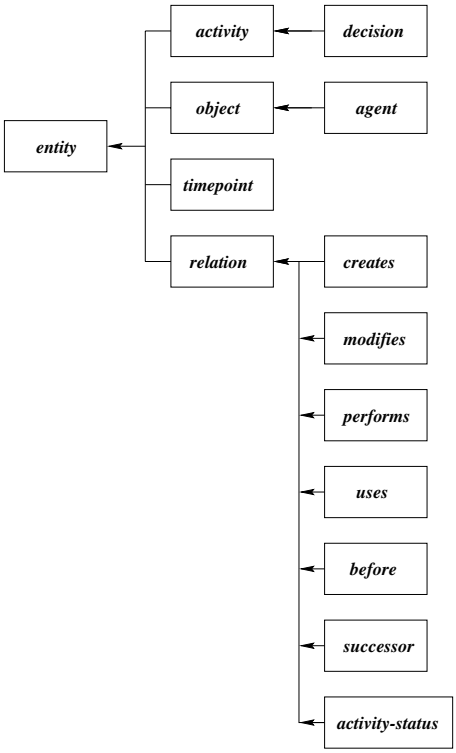

Fig. 2.2. The PIF class hierarchy

Figure 2.2 shows the PIF class hierarchy [62]. Modelling concepts such as *entity, activity, object, agent* and the notion of *time* are captured as PIF classes in the hierarchy. Everything in PIF is a subclass of the root class *entity*. There are four subclasses of *entity*: *activity, object, timepoint* and *relation*. Each subclass may also have its own subclasses. A subclass is indicated by an outgoing arrow from itself to the corresponding superclass. This relationship between the subclass and the superclass is a *specialisation* relationship. For instance, in PIF, *decision* (making) is a special type of *activity*, and *before* describes a particular kind of (temporal) *relation*.

The benefits of allocating all modelling concepts in a PIF class hierarchy with the property of *inheritance* is similar to that of the *Process Handbook Project*. The conceptual entities are clearly identified and relationships between them specified. New concepts can be added to the hierarchy and fundamental properties can be automatically inherited from their superclasses. This speeds up the process of creating, manipulating and evaluating process modelling concepts.

In addition to the concept of *process specialisation*, the PIF class hierarchy has also been adapted in our formal work to suit BSDM's business modelling method, which enables our work to enjoy all of the above benefits of PIF. Moreover, because we have taken an approach similar to PIF, we enabled BSDM's business models to more easily communicate with any other process languages communicating through PIF. It also enables BSDM models to be translated to other process languages through PIF which already is linked to many other languages.

2.5 Review of Existing Enterprise Modelling Tools

The field of enterprise modelling, especially business process modelling and workflow, is a very active area during the past five years in which new research as well as commercial tools have been built rapidly. This section therefore is not intended to provide a comprehensive review, but discusses example tools currently available and their characteristics. From what has been collected, most modelling tools provide quality related support for some aspects of modelling activities. Based on their functions, such tools are divided into two categories.

The first category of tools provide primarily capturing and report-generating functions for specific modelling methods. Examples of such tools are *RBPL* [108] which provides its own business process modelling language as well as the corresponding documentation facilities; Paradigm Plus [84] supports various modelling methods, such as *Booch* [10], *OMT* [93] and *UML* [11], and exporting facilities for these methods; *BP WIN* [66] provides drawing and report-generation facilities for IDEF0 [80] and *Data Flow Diagrams (DFD)*; *AI0 Win* [60] supports the drawing and documentation of *IDEF0* models and can export its details to other tools, such as *ProSim* [60]; *Win A&D* (or *Mac A&D*) [35] provides documentation and reporting facilities for various modelling languages, such as the *Class Model*, *Entity Relational* data model and *Data Flow Dia-*

grams; another relevant business process modelling tool is *Procedural Builder* [7] developed by AIAI which allows the user to build an adapted version of *IDEF3* models [72] and can communicate with the *Enterprise Toolset* that is supported by knowledge described in the *Enterprise Ontology* [6].

In addition to providing documentation and report-generation functions, the second category of tools also provide simulation (and/or process execution) facilities for the described models. For example, *ProSim/ProCap* can simulate its own processes and can import processes that have been drawn using other tools, e.g. *AI0 Win* and *Visio* [76]. *Simprocess* [17] is an object-oriented process modelling and analysis tool based on its own simulation language *Simscript* for analysing complex, dynamic systems. *BPSimulator* [110] is a discrete event simulation tool which simulates business processes using statistical simulation methods. *iThink* [47] is a tool for simulating system dynamic models – these system dynamic models were initially designed to simulate physical systems, e.g. a fluid system – this technology has been used by *iThink* to model the flow of a business environment. ARIS Toolset [54] provides its own modelling language which supports *Activity-Based Costing (ABC)* and *Balanced Score Card (BSC)* to record and analyse a business performance. Yu, Mylopoulos and Lesperance [118] capture the actors, actors' goals and dependencies between them in a business operation in *Strategic Dependency Models* and *Strategic Rationale Models* which allow simulation of business processes to be carried out and opportunities, vulnerabilities and patterns of dependencies to be explored. SAP R/3 [97] offers a client/server architecture and distributed open system solution whose in-house business processes are under-pinned by the modelling language *EPC (Event-driven Process Chain)*. Other recent work are to provide a workflow system that supports business process definitions and execution, e.g. Oracle Workflow [81] and Staffware [104].

The simulation support given by the tools in the second category is largely of the type that is usually found in performance studies. It allows the user to specify type and frequency of business processes and the company resources required by these processes. Running such simulations can help identify bottlenecks in the company's operations, but also means that the user has to specify numerous input parameters.

In general, there is very little, if any, exploitation of the rich contextual knowledge that is implicit in the models that have been captured through the corresponding documentation features of the above tools. One reason for this is that there is no underlying mechanism to allow such knowledge to be built into and used by these tools. Consequently, they are unable to provide modelling support beyond that based on a model syntax.

Together, KBST-BM and GMA support the basic modelling activities such as drawing, documenting, navigating, summarising and reporting, but they also provide support related to model semantics, such as consistency checking, error-correction advice-giving, alternative visualisation of the model (some based on derived information), simulation of processes, model building and refining guidance (by referring to and comparing with standard or existing models), model verification and validation, and model reuse. In particular, much of the

model quality checking work can be automated to such an extent to make it feasible and effective to do so in an applied context.[10] More details about KBST-BM and GMA are given in Chapter 5, 6, 7, 8 and 9.

2.6 Exercises

1. What is an Enterprise Modelling method? Why Enterprise Modelling?
2. What are the different types of Enterprise Modelling (EM) methods? Explain BSDM's relation with EM.
3. What is the relationship between BSDM and generic software development methods in Software Engineering? Can BSDM be useful as a part of software development processes?
4. What are the main concepts in a (business) process model?
5. Describe the process interchange language of PIF, and discuss whether it is useful to have such a language.

[10] KBST-BM and its successor KBST-EM have been used in research projects AOEM [56], AKT [1] and AIAI commercial projects.