Collaborative Modeling Using UML and Business Process Simulation

Sergio de Cesare and Alan Serrano
School of Information Systems, Computing and Mathematics
Brunel University
Uxbridge, Middlesex, UB8 3PH, United Kingdom
{sergio.decesare|alan.edwin.serrano-rico}@brunel.ac.uk

ABSTRACT

Business process (BP) change projects often involve the redesign of organizational information systems (IS). To successfully align the design of processes and IS, collaboration between BP and IS analysts is required. This is usually done with the aid of IS and BP models. This paper argues that such models often fail to address the concerns and information needs of both groups because they are not created in a collaborative way. Thus, it proposes a modeling framework that maps the constructs used in IS models to those used in BP models so that both groups consider their organizational views in their designs. The framework was applied in a business process change project for a multinational pharmaceutical organization. The results of this case study indicated that: a) sharing the information between BP and IS models contributed to foster collaboration between analysts in both domains; and b) aiding the collaboration between analysts produced more integrated solutions.

Keywords: Information systems design, business process design, simulation, modeling, business process modeling.

1. Introduction

The view that Business Processes (BP) and Information Technology (IT) are closely related is not new and has been widely documented by professionals and academics. For example, Childe et al. [3] state that the initiative to move towards Business Process Reengineering (BPR) in many cases originates from the IT departments. Similarly, Davenport and Short [4] affirm that, when used together, IT and BPR have the potential to create a new type of industrial engineering. Rhodes [5] supports the idea that IT must not be viewed just as a BP supporting tool, but as a management philosophy. Turban et al. [6] assure that many organizations rely heavily on Information Systems (IS) to achieve their business goals and objectives and to gain or retain competitive advantage. Therefore, many Business Process Change (BPC) projects usually advocate the generation of new IT or propose changes to the existing one.

To design business processes that satisfy in the best possible way the organizational objectives, a number of design approaches have been proposed, most of them relying on a wide spectrum of modeling techniques. Many process approaches acknowledge that the use of a new IT infrastructure may influence the design of business processes and advocate the idea that the design of BP and IT should be coordinated. Furthermore, initial stages required in most of business process change approaches are closely related to those phases used for IS design. To illustrate this point, Davenport's [1] framework for process innovation and Information Engineering (IE) systems development methodologies are briefly described and the similarities between those approaches are drawn next.

Davenport's [1] framework for process innovation consists of the following five steps (see Fig 1):

a) Identifying process for innovation: The first step of this framework detects the processes performed by
the organization in order to select those that are more suitable for redesign.

b) Identifying change levers: The second step highlights the importance of identifying enablers of process innovation. The author distinguishes three major enablers: information technology enablers, organizational enablers, and human resource enablers.

c) Developing process visions: Consists of providing the necessary ‘linkage’ between strategy and action, specifying measurable objectives and attributes of the future process state.

d) Understanding existing processes: The fourth step helps to facilitate communication among participants, to facilitate migration and implementation planning, to ensure that recognized problems are not repeated in the new process, and finally to provide a measure of the value of the proposed innovation.

e) Designing and prototyping the new processes: According to the results of step four, it is necessary to propose a number of alternative re-designed processes. These processes are compared and evaluated in the fifth step in order to decide a specific solution. In this step, a migration strategy towards the selected solution is formulated, followed by the implementation of the new organizational structures and systems.

Information Engineering (IE) is a methodology that covers all aspects of the life cycle and is viewed as a framework that comprises a variety of techniques that are used to develop good quality IS in an efficient way. Although there are some differences of opinion as to the origins of IE, the views of Martin [7], Finkelstein [8], and Avison and Fitzgerald [9] will be included here. IE follows a top-down approach to systems development. It begins with a top-management overview of the organization in order to ensure the coordination of other systems. The IE methodology is divided into four levels:

1. Information strategy planning level: It is concerned with the identification of business goals and the articulation of ways technology can be used to achieve these goals.
2. Business area analysis level: It is concerned with developing business models that represent organizational activities and their data requirements.
3. System planning and design level: It is concerned with matching the behavior of the system with the user requirements and with the extant technology.
4. Construction and cutover level: Its major objective is to build and implement the system as specified in the three previous levels.

As it can be derived from these two examples, the initial phases undertaken in a BPC study are very similar to those used in an IS design process. For example, steps a and b of Davenport’s framework follow similar objectives to level 1 of the IE methodology. Both are looking at the understanding of the way the organization operates and the ways IT can be used to improve the organization’s performance. Similarly, the development of business process models is performed in step e of Davenport’s framework as well as in level 2 of IE.

<table>
<thead>
<tr>
<th>BP Modeling Techniques</th>
<th>Functional</th>
<th>Behavioural</th>
<th>Organisational</th>
<th>Informational</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flowcharting</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Limited</td>
</tr>
<tr>
<td>IDEF0</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Limited</td>
</tr>
<tr>
<td>IDEF3</td>
<td>Limited</td>
<td>Limited</td>
<td>No</td>
<td>Limited</td>
</tr>
<tr>
<td>Petri Nets</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Discrete Event Simulation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>System Dynamics</td>
<td>Limited</td>
<td>Yes</td>
<td>Yes</td>
<td>Limited</td>
</tr>
<tr>
<td>Knowledge-based Techniques</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Role Activity Diagramming</td>
<td>No</td>
<td>Limited</td>
<td>Yes</td>
<td>No</td>
</tr>
</tbody>
</table>

Table 1. BP Modeling Techniques Vs Process Modeling Perspectives (adapted from [2])

The fact that both BPC and IS projects pursue similar objectives in their initial phases, suggests that the modeling mechanisms used in these approaches encapsulate information that is common to both domains. Furthermore, it can be seen that approaches in both domains tend to consider the system as the sum of social, organizational and technological parts. Therefore, such approaches imply the collaboration between stakeholders in these areas. Despite these facts modeling mechanisms used in BP and IS projects tend to be either BP or IS oriented, thus, usually failing to foster collaboration between these domains and more importantly, failing to model an integrated view.

To address this problem, this paper proposes a framework that maps the constructs used in IS models to those used by BP models. The benefits of doing so can be seen in different ways. First, mapping constructs among models in both domains helps analysts to better understand the views depicted in the other domain’s models. Second, using the framework to derive BP models from IS models can help to foster collaboration between analysts. In addition, this also helps to incorporate IS views into BP models, hence to produce more integrated solutions. Finally, the time taken to design process models can be reduced.

The mapping framework is based on business process simulation (BPS) and the Unified Modeling Language (UML). Simulation is proposed because it is argued to be successful when using it for communication among system stakeholders. In addition, BPS modeling supports the creation of dynamic models of organizational processes and information systems. Dynamic models offer more opportunities for evaluating business processes and information systems strategies.

The paper is structured in the following way. Sections 2 and 3 describe some of the modeling techniques used in both BP and IS domains to depict
business processes, highlighting the advantages that simulation has over other modeling techniques. This will be used to a) justify the selection of BPS for the mapping framework proposed in subsequent sections, b) identify the information needed to derive such models, and c) to identify the information conveyed by these models and that can be used interchangeably in both UML and BPS models. Section 4 describes the proposed modeling framework that will be used to derive BPS models from IS models and vice versa. Section 5 is a description of the case study used to evaluate the framework previously described. Section 6 presents the application of such a framework to the case study. Finally section 7 draws conclusions.

2. Modeling the Organization: The BPC Perspective

A business process model must be capable of providing various information elements to its users. Such elements include, for example, what activities comprise the process, who is performing these activities, when and where these activities are performed, how and why they are executed, and what data elements they manipulate. Business process modeling techniques differ in the extent to which their constructs highlight the information that answers these questions. To provide this information, a modeling technique should be capable of representing one or more of the following modeling perspectives [10]:

- Functional perspective: Represents what process elements (activities) are being performed.
- Behavioral perspective: Represents when activities are performed (for example, sequencing), as well as aspects of how they are performed through feedback loops, iteration, decision-making conditions, entry and exit criteria, and so on.
- Organizational perspective: Represents where and by whom activities are performed, the physical communication mechanisms used for transfer of entities, and the physical media and locations used for storing entities.
- Informational perspective: Represents the informational entities (data) produced or manipulated by a process and their relationships.

Business process modeling techniques are used to develop models of the way an organization operates, the processes the organization has and how they are related with different organizational areas in order to give an understanding of possible scenarios for improvement [11, 12]. Flowcharting, IDEF0, IDEF3, Petri Nets, System Dynamics, Knowledge-based Techniques, Role Activity Diagrams, Activity Based Costing, and Business Process Simulation are some examples of BP modeling techniques widely used in the organizational domain. Table 1 compares the most popular BP modeling techniques in terms of Curtis et al.’s [10] process modeling perspectives. According to Giaglis [2] only business process simulation (also known as discrete-event BPS) is able to address fully three of the process perspectives.

Many researchers advocate the idea that business process simulation is one of the most suitable modeling techniques that can be applied to the business process domain. Existing literature in this domain provides an extensive list of reasons to support this argument. Among them the most relevant are:

- Simulation modeling supports the creation of dynamic models of organizational processes and information systems [13, 14]. Dynamic models offer more opportunities for evaluating business processes and information systems [15].
- The visual interactive features of many simulation packages allow multidisciplinary team members to understand the model and communicate about it [15, 16].
- Besides capturing the processes, simulation also supports the quantitative evaluation of different alternatives before implementation [15]. Quantitative process metrics that can be addressed by simulation include, but are not limited to, costs, cycle time, serviceability, and resource utilization [17, 18]. These metrics also form the basis for evaluating alternatives in BPR [2, 19].

3. Modeling the Organization: The IS Perspective

Models in the IS domain can be used to represent many different aspects of the IS process, hence the wide spectrum of modeling techniques. The following paragraphs describe some of the different aspects from which an organization can be modeled.

Enterprise modeling deals with understanding the way the organization operates. To do so, it is necessary to capture the organization’s structure, the business rules that govern and affect the organization’s operations, the goals, tasks and responsibilities of the organization’s members, and the data generated and manipulated by the organization.

Data modeling deals with the manipulation and management of the information generated and used by the organization. Traditional techniques such as entity relationship diagrams could be used to capture these requirements. Object-oriented techniques such as class diagrams, however, are increasingly replacing traditional techniques.

Behavioral modeling deals with modeling the dynamic behavior of the current and the required system. In this respect it is necessary to model both the way the
stakeholders use the current system (manual and IT systems) and the system itself. Structured and object-oriented methods provide a variety of methods with different levels of abstraction to model current and required systems.

Domain modeling provides an abstract description of the environment in which the required system will operate. This type of model can provide a detailed understanding of the domain and an opportunity for requirement reuse within a domain.

Most of the IS modeling techniques specialize in different aspects, depending on the stage at which they are applied. For example they can be used to understand either the overall function of the system in question, to understand IS data structures, or to model the processes involved in the IS. Table 2 shows a classification of modeling techniques (adapted from [9]) according to the stage in which they can be applied and the aspect they address.

Rich pictures, conceptual models, data flow diagrams (DFD), entity relationship diagrams (ERD), state transition diagrams, IDEF1x, and object-oriented (OO) diagrams can be mentioned as the most dominant IS modeling techniques [20]. The remainder of the paper will focus only on use case and activity diagrams of the UML.

![Image](image-url)

Table 2 Classification of Modeling Techniques (adapted from [9])

<table>
<thead>
<tr>
<th>Strategy</th>
<th>Jigsaw</th>
<th>Aspects addressed</th>
<th>Domain/Process</th>
</tr>
</thead>
<tbody>
<tr>
<td>Investigation &amp; Analysis</td>
<td>Rich pictures</td>
<td>Rich pictures, objects, structure diagrams, use cases</td>
<td>Entity modelling, data flow diagrams, entity life cycle, decision trees, decision tables, action diagrams, root definitions, conceptual models (UML)</td>
</tr>
<tr>
<td>Logical design</td>
<td>Matrices</td>
<td>Matrices, structure diagrams</td>
<td>Normalisation, entity modelling, data flow diagrams, entity life cycle, decision trees, decision tables, action diagrams</td>
</tr>
<tr>
<td>Implementation</td>
<td>Matrices</td>
<td>Matrices, structure diagrams</td>
<td>Normalisation, entity modelling, data flow diagrams, entity life cycle, decision trees, decision tables, action diagrams</td>
</tr>
</tbody>
</table>

4. Mapping BP and IS Models: Rationale

From the previous discussion of business process modeling techniques, simulation was positioned as one of the most popular and complete modeling tools in this domain. This is mainly due to the differences between dynamic and static models. A static simulation model is a representation of a system at a specific point in time or when time simply plays no role. A dynamic simulation model, however, aims to represent a system as it evolves over time. Therefore, BPS models have the ability of describing processes in the organizational system with graphical symbols, as static models do, and provide quantitative information about these processes.

Similarly, from the analysis of IS modeling mechanisms, UML’s modeling techniques were considered to be very comprehensive when modeling IS. The different modeling techniques used in UML (e.g. use cases, activity diagrams and class diagrams) together with the fact that these techniques are widely used by the IS community suggest that UML can be proved useful for the purposes of this paper. Therefore, this paper uses these techniques to illustrate that BPS and IS models convey similar information, thus, they can be used as the foundation to create BPS from IS models, or vice versa. Furthermore, the creation of such models will enable practitioners in both domains to integrate IS perspectives into BP models and vice versa.

The following is a discussion of the main elements needed to create BPS models and possible ways this information can be obtained from IS models. The main components to construct business process simulation models can be described as follows:

- **Processes:** Defined as the series of activities needed to achieve a specific goal.
- **Activities:** Activities use up resources to produce products and services. Activities combine to form business processes.
- **Entities:** Entities are objects that circulate through the model. They represent things (e.g., parts, deliveries, and people) and information (orders, service requests, etc.) that flow from activity to activity.
- **Resources:** A resource is an agent that is required to perform the tasks associated with an activity. People, equipment, vehicles, money, and space can be modeled as resources. The limited availability of resources is an important constraining factor in business processes.

Entity components, though, have some characteristics that make a major difference between dynamic and static models. Entities can have passive or active states. An active state frequently involves the cooperation with other entities and can usually be forecasted by taking an appropriate probability distribution if the model is stochastic. For example, a salesman is engaged during a random period of time to process an order (‘take an order’ activity). In this example the entities salesman and order are cooperating to perform the ‘take an order’ activity. A passive state, or queuing state, involves no cooperation between different entities and it is generally a state in which the entity waits for something to happen. For example, the time an order is waiting to be processed while the salesman is engaged processing other orders. The period of time an entity will spend on a queue cannot be determined in advance because it depends on the duration of the preceding and succeeding activities.

Therefore, apart from the static simulation elements identified before, dynamic information such as the probability distributions needed to recreate the cooperation with other entities needs to be collected.
This paper advocates the idea that IS models can be used to obtain much of the information needed to build BPS models. For instance, UML’s business and system use cases can be mapped to define business processes and the activities within those processes. The actors in system use cases are a good representation of the entities used in business processes simulation model. This information can also be complemented by looking at class diagrams. More importantly, system use cases are able to represent time-related constraints of the system that can be used to propose appropriate probability distributions within the business processes and/or activities.

The following section proposes possible mappings between business process simulation modeling and UML modeling.

4.1. Mapping Framework

The Unified Modeling Language was originally conceived as a general-purpose language for representing software systems. The UML is defined as “a language for specifying, visualizing, constructing, and documenting the artifacts of software systems, as well as for business modeling and other non software systems” [21]. The claim of general applicability (i.e., to model both software and business systems) has been debated and subject to criticism in recent years. As a consequence, fruitful results have been produced in trying to extend the UML, in terms of modeling elements and patterns, aimed at increasing the language’s fit to modeling organizational phenomena. In this sense, the latest UML 2.0 version has adopted some of the suggestions that over recent years both academics and practitioners have proposed [22].

The UML defines a wide number of diagram types. This paper will only focus on use case and activity diagrams. The reasons for this are three-fold. Firstly, both use cases and activity diagrams have been recognized by the literature as having a role in modeling organizations and their processes. Secondly, business process simulation is intimately related with activity diagrams. Both in fact can be derived from Petri Nets, a more generalized version of flow diagrams. Thirdly, although other diagram types (e.g., class, interaction and state diagrams) can play a part in business modeling, an in-depth discussion of all diagrams would go beyond the scope of this paper.

The main modeling elements that make up any process simulation model are: processes, activities, entities and resources. In addition simulation also necessitates of whole-part hierarchies (e.g., between processes and activities) and distribution/time constraints. These are the fundamental elements required to represent a business process simulation and are common across the simulation techniques documented in the literature. Counterparts to these modeling elements can be identified in both UML use case and activity diagrams. In the subsections that follow, mappings between use cases and activity diagrams to BPS will be discussed. The notation will be presented in sections 4.1.1 and 4.1.2. The mapping framework will then be applied through a case study example in Section 6.

4.1.1 Use Cases and BPS

A use case can be defined as a description of a set of sequence of actions, including variants, that a system performs that yields an observable result of value to a particular actor. The term ‘observable’ refers to the outcome of a process. In a business context it is plausible to assume that a use case models a business process which is identifiable in terms of a service (observable outcome) that an actor expects. The relationship with business simulation is in the elements of a use case. Table 3 highlights the relationships which at this stage can be considered as the first step in identifying the potential mappings between business process simulation and use cases.

<table>
<thead>
<tr>
<th>Use Case Elements</th>
<th>BPS Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description of a set of sequence of actions</td>
<td>process</td>
</tr>
<tr>
<td>Action</td>
<td>activity</td>
</tr>
<tr>
<td>Observable result</td>
<td>goal</td>
</tr>
<tr>
<td>Actor</td>
<td>resource</td>
</tr>
</tbody>
</table>

Table 3. First-cut mapping between use cases and BPS

The definition of use case does not provide any explicit reference to whole-part hierarchies. In use case diagrams, such hierarchies can be represented through «include» and «extend» relationships between use cases. Include relationships model mandatory participation between use cases, whereas extend relationships model optional or conditional participation.

The preliminary conclusion to draw is that in terms of basic modeling elements both use cases and BPS draw on a common set of elements. The following observations can however be made: (1) As Table 3 illustrates, the concept of entity is not explicit in the definition of use case; (2) A use case is predominantly a textual representation of a service/process and the interactions between the actor and the system; and (3) Although non-functional requirements can be represented (e.g., time constraints), these too are described in natural language.

BPS however requires precise and formal methods of representation which natural language cannot effectively satisfy. As a partial solution, a structured form of natural language can be adopted, but this solution also lacks in totally resolving the problem. Furthermore, simulations are dynamic in nature; hence require a formalism
(notation) which enables the modeler to ‘run’ the simulation, preferably in an automated way. Graphical notations achieve this purpose and can help in overcoming the deficiencies of use cases in relation to process simulation. However, use cases should not be discarded altogether, given that they are a useful tool in capturing business and system requirements as well as providing the premise for capturing and detailing the same requirements in a more formal and structured manner. In adopting the UML for business process simulation it is necessary to integrate the application of use cases with a graphical technique, namely activity diagrams. Table 4 summarizes the further mappings described above.

<table>
<thead>
<tr>
<th>Use Case Elements</th>
<th>BPS Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Description of a set of sequence of actions</td>
<td>process</td>
</tr>
<tr>
<td>Action</td>
<td>activity</td>
</tr>
<tr>
<td>Observable result</td>
<td>goal</td>
</tr>
<tr>
<td>Actor</td>
<td>resource</td>
</tr>
<tr>
<td>«include» and «extend» relationships</td>
<td>whole-part hierarchies</td>
</tr>
<tr>
<td>Non-functional specifications documented in natural language</td>
<td>time/distribution constraints</td>
</tr>
</tbody>
</table>

Table 4. Mapping between use cases and BPS

4.1.2 Activity Diagrams and BPS

Activity diagrams provide natural support for expressing textual use case descriptions in a graphical and structured manner. Activity diagrams support the representation of both control and data flows, which are essential for BPS. In a BPS model the participation of entities is essential to process simulation. These entities can be either inputs or outputs to activities, hence data flows. Furthermore, entities can change state, sometimes even generating or destroying other entities.

In the UML, activity diagrams model activities. An activity specifies the coordination of executions of subordinate behaviors, using a control and data flow model. Subordinate behaviors can be defined in terms of other activities or actions. An action is the fundamental unit of executable functionality in an activity. Actions can be coordinated and linked to other actions through a flow of control and can also be linked to objects modeling data flow at the same time. An action can have one or more incoming flows and will not fire until each of the incoming flows is triggered. This allows for actions to be executed simultaneously and synchronized between them. Table 5 provides basic mappings between activity diagrams and BPS. These mappings will be exemplified in section 6.

<table>
<thead>
<tr>
<th>Activity Diagrams Elements</th>
<th>BPS Elements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>Activity</td>
</tr>
<tr>
<td>Activity</td>
<td>Process</td>
</tr>
<tr>
<td>Action-activity relationship</td>
<td>Whole-part hierarchy</td>
</tr>
<tr>
<td>Activity-activity relationship</td>
<td>Whole-part hierarchy</td>
</tr>
<tr>
<td>Partition</td>
<td>Resources</td>
</tr>
<tr>
<td>Objects</td>
<td>Entities</td>
</tr>
<tr>
<td>Time signal/Synchronization bars</td>
<td>Time</td>
</tr>
</tbody>
</table>

Table 5. Mapping Between Activity diagrams and BPS

5. Case Study

Two organizations are considered in the case study presented here to demonstrate the viability of the proposed framework. One company is a branch of a multinational pharmaceuticals organization (referred to as Org-A), while the other is a small organization that distributes Org-A’s products (referred to as ‘Org-B’). Org-A opened a production site in order to manufacture and market products for South Europe and the Middle East. The headquarters are in Athens; the plant and the warehouse are located in Mandra, a suburb of Athens; a smaller office responsible for managing Org-A sales for northern Greece in Thessaloniki, the second largest city in Greece.
The case study analyzed in this section was carried out in Org-A’s medical division. This division deals with hospital goods such as surgical dressings, disposable surgical packs, and medical devices like blood glucose and monitoring systems. The medical division has as major customers large public sector organizations like hospitals, health care organizations, networks of physicians, and the government. The commercial business unit deals with other business customers that sell Org-A products to end consumers (for example chemists and supermarkets). The commercial division is involved in producing and marketing the products. The medical division does not produce goods but instead imports them from other Org-A production sites in Europe and these are stored in the Mandra warehouse. This means that the warehouse is the departure point for all the products that are distributed to the company’s customers through a variety of distributors. One of these distributors is Org-B.

Org-B has an agreement with Org-A according to which it is the exclusive distributor of Medical unit products for northern Greece. The agreement signed between both companies states, among other issues, that Org-B is responsible for:

a) Receiving orders from Org-A customers.

b) Maintaining an adequate inventory of products that fulfill the orders.

c) Distributing the ordered products to customer premises.

Org-A has its own site in Thessaloniki consisting of a small team of salesmen and office staff. The salesmen are responsible for informing customers about Org-A’s variety of products, obtaining and preserving company customers, and signing sales contracts.

5.1. Identified Problems

Org-A and Org-B managers identified, after a series of discussions and a brief analysis of the current processes, the following problems:

a) Excessive order lead times. Orders are fulfilled under lead times that were far higher than the targets specified by both organizations. Moreover, the levels of stock are not appropriately maintained. This in turn produces the request of backorders, which contribute even more to the order fulfillment time.

b) Inappropriate inventory replacement policy. In order to solve the problem of having a great number of backorders, the Org-A warehouse managers implemented a generous replenishment policy for Org-B’s warehouse. This policy, however, causes other problems. For instance, it causes considerable warehouse holding costs for Org-B. Similarly, this policy also causes problems related with sensitive medical products reaching their expiry date.

c) Poor customer service. A considerable number of customer complaints, regarding delivery delays.

d) Excessive invoice lead times. Invoices take too long to reach customers, which in turn causes poor cash-to-cash cycle for both organizations.

e) Data and work duplication. Org-B and Org-A use different warehouse software to monitor their stocks. Org-A needs to know the stock levels at Org-B in order to plan and schedule replenishment shipments. Therefore, Org-B’s stock levels are sent to Org-A. Three major problems can be identified from this scenario. First, data duplication between both companies. Secondly, duplication of work (double typing the same information). Finally, some mismatch between the data reported in Org-B and Org-A databases was reported.

f) Information sharing. Org-B and Org-A information infrastructures are incompatible. Consequently, the companies have to rely on paper forms to share information. This causes the duplication of data and effort, which in turn produces slow processing times.

To resolve these problems a business process simulation exercise was performed. The simulation model derived from this exercise will be used in the following section as a case study to demonstrate the feasibility of constructing BPS models from UML models using the guidelines proposed in this paper.

6. Mapping between Models

This section applies the guidelines presented in section 4.1 to the case study previously presented. The example will highlight the process of translation from UML use case and activity models to BPS models. This process can be summarized as follows:

1. Identify the business processes represented in the use case diagrams. This step produces a holistic view of the modeled processes with all associated elements listed in Table 4.

2. Map the activities and actions to business processes and define the corresponding whole-part hierarchies.

3. Recognize time constraints in the activity diagrams and represent them in the BPS models.

The UML models used in this example are illustrated in Figures 2, 3, 4 and 5. Figure 2 represents the use cases of the problem domain. Figure 2 describes the processes and activities carried out when a customer places an order. This gives rise to a hierarchy of processes. The main process is ‘Input Customer Order’ and the two sub processes are ‘Issue Backorder’ and ‘Issue Order’. According to the mapping guidelines the three processes can be represented as use cases. The whole-part hierarchy
is mapped to either «include» or «extend» relationships depending on its characteristics.

Figure 2: Use Case Diagram

Figure 3. Input Customer Order Activity Diagram

Figure 4. Issue Backorder Activity Diagram

Figure 5: Issue Customer Order Activity Diagram

6.1. Mapping Use Cases to BPS Models

The use case diagram provides the basis for organizing the BPS model. Each use case can be considered as a business process to simulate. Three use cases have been taken into consideration and represented as processes the BPS models of Figures 6, 7 and 8. The actors are represented as resources. For example, the resources allocated to the ‘input order details’ activity (Figure 6) are employees of Org-A and Org-B. The same procedure is applicable to the two subprocesses.

The use case diagram provides a holistic view of the whole system in which the processes to be simulated are represented. The textual descriptions are then represented by activity diagrams whose elements can be mapped to BPS models.

6.2. Mapping Activity Diagrams to BPS Models

The three activity diagrams (or activities in UML terms) have been mapped to business processes. The result of this mapping is illustrated in figures 6, 7 and 8. Obvious mappings include UML actions to BPS activities and the representation of synchronization between activities. Less obvious and implicit in the diagrams themselves is the whole-part hierarchy. In activity diagrams this is achieved through actions which refer to activity diagrams. This is the case, for example, of the ‘issue order’ and ‘issue back order’ actions in the ‘Input Customer Order’ activity diagram.

As shown in Figures 3 and 5, objects (e.g., ‘Order’ and ‘Backorder’) and object flow are modeled as inputs and/or outputs to actions. This feature is not used much by UML practitioners, but becomes essential when wanting to represent the objects that are created,
transformed and destroyed by actions/activities. It is possible to note that such objects are not clearly represented in the BPS models. This represents a limitation of the specific modeling tool/language adopted.

The use case actors have been mapped to partitions or swimlanes providing a clear-cut division of responsibilities between them. As previously mentioned, the actors (partitions in this case) map to BPS resources.

7. Conclusions

This paper has presented a framework for mapping business process simulation models to UML models. This framework has been formulated to support the following claims: (1) BPS and IS models share common information; (2) The information conveyed in UML models can be used to create BPS models and vice versa; and (3) (Re)using IS models to derive BPS models (and vice-versa) can reduce modeling time, and more importantly, promote BP and IS modeling integration. Such integration is achieved thanks to a mechanism for translating the modeling terminology and techniques of both groups of practitioners.

To support these claims, this paper uses existing literature in BP and IS domains to highlight the relationships between BP and IS modeling domains. Based on these findings this paper proposed a modeling framework to derive BPS models from UML models and applied it in a case study.

The application of the framework in the case study provides evidence to support the claim that the creation of BPS models could be simplified, hence, reduce modeling time. A more significant advantage, however, is related to the way in which the application of this framework can aid practitioners to have a deeper understanding of the relationships between BP and IS. For instance, to derive BPS models from UML models, as proposed in the framework, IS practitioners need to first understand information systems models in more detail. The proper analysis of IS models will necessarily lead IS practitioners to a deeper understanding of the way existing IS support business processes. Furthermore, the literature review found that one major advantage of BPS over other modeling techniques is that it supports the quantitative evaluation of different BP alternatives before implementation. Since the BPS models produced by the implementation of the framework are derived from IS models, the development of alternative BP scenarios can now be carried out considering the effects that alternative IS designs may have on the processes. This can also be true for IS design. IS design options can be better tailored to satisfy the business process demands identified with the aid of the simulation model. Thus, it can be said that the creation of BPS models from UML models (and vice versa) contributes towards a modeling environment that better reflects the integration of BP and IT.
References


