

# Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems\*

Jose Luis de la Vara, Juan Sánchez, and Óscar Pastor

Department of Information Systems and Computation, Technical University of Valencia,  
Camino de Vera s/n, 46022, Valencia, Spain  
{jdelavara, jsanchez, opastor}@dsic.upv.es

**Abstract.** Although requirements analysis is acknowledged as a critical success factor of information system development for organizations, problems related to the requirements stage are frequent. Some of these problems are lack of understanding of the business by system analysts, lack of focus on the purpose of the system, and miscommunication between business people and system analysts. As a result, an information system may not fulfil organizational needs. To try to prevent these problems, this paper describes an approach based on business process modelling and purpose analysis through BPMN and the goal/strategy Map approach. The business environment is modelled in the form of business process diagrams. The diagrams are validated by end-users, and the purpose of the system is then analyzed in order to agree on the effect that the information system should have on the business processes. Finally, requirements are specified by means of the description of the business process tasks to be supported by the system.

**Keywords:** Business process modelling, system purpose, BPMN, Map, task description.

## 1 Introduction

Requirements analysis has been widely acknowledged as a critical success factor of software projects [31]. If not properly addressed, requirements can cause a project to fail. Nevertheless, practical experience proves that problems can easily arise from the requirements stage of information system (IS) development for organizations. Some of these problems are lack of understanding of the business by system analysts, lack of focus on the purpose of the system, and miscommunication between business people and system analysts. Since these problems can hinder business/IT alignment [20][25], the IS does not fulfil organizational needs.

Requirements must be defined in terms of phenomena that occur in the business environment [36]. However, it is common for requirements documentation to be solution-oriented, to not reflect the business environment, or to only consist of a data model in the

---

\* This work has been developed with the support of the Ministry of Education and Science of Spain under the project SESAMO TIN2007-62894 and the program FPU, and cofinanced by FEDER.

form of a class or entity-relationship diagram. As a solution, the requirements engineering community has acknowledged the importance of the use of business concerns to drive requirements elicitation [29]. More specifically, the importance of organizational modelling during requirements analysis [18] and the role that system analysts must play as business analysts [16] have both been acknowledged. Organizational models depict the structure and behaviour of an organization and are very useful in helping system analysts to properly understand the business environment and the system requirements. Among other approaches, business process modelling has been declared as a good approach for organizational modelling and also as a must for IS development [2][13].

Nevertheless, business process models are not always enough to analyze the business context. Organizations often decide to introduce or modify an IS to solve a specific problem or need. These problems or needs correspond to the goal that the system must fulfil, i.e., the system purpose. Therefore, it is important for system analysts to explore the goals of different stakeholders and the activities that they carry out so that they can define purposeful requirements [26]. When the system purpose is not very complex, it can be directly analyzed on business processes, but sometimes it requires a deeper analysis. In the latter case, the use of a goal-driven approach that facilitates purpose analysis helps to better understand the system purpose and, consequently, to better respond to end-user needs.

Furthermore, good communication between business people and system analysts is essential at the requirements stage [17]. However, it can be difficult to achieve because of the existence of a gap between business and computing domains [32] that can cause mismatches between what end-users say and what system analysts understand. One reason for miscommunication is that the requirements models used to interact can be hard to understand and validate by end-users because of their lack of computing background. Therefore, models that facilitate communication during requirements analysis should be used. According to the software community [8] and to cognitive experiments [1], the use of process models when developing an IS can make human understanding and communication easier, and can help interaction between system analysts and end-users.

This paper presents a requirements analysis approach based on business process modelling and purpose analysis in order to try to prevent the problems described above. It is the result of a project between the Technical University of Valencia and the company CARE Technologies (<http://www.care-t.com>). The approach is characterized by the use of business processes for organizational modelling and as starting point of requirements analysis, the focus on system purpose, the use of BPMN [23] for business process modelling, the use of the goal/strategy Map approach [27] for system purpose analysis, the detailed specification of functional requirements, and the involvement of end-users. This involvement is very positive when modelling an organization [30] and is a success factor in software development [31].

Organizations are modelled in the form of business process diagrams. The diagrams are validated by end-users, and the system purpose is then analyzed in order to come on agreement on the effect that the IS should have on the business processes. Finally, requirements are specified by means of the description of the business process tasks to be supported by the IS. These well-defined requirements are the input of the subsequent development stages.

The paper is organized as follows: section 2 describes the notations of the approach; section 3 describes the case study; section 4 presents the description of the approach; subsections 4.1 and 4.2 describe purpose analysis and functional requirements specification in detail; section 5 describes the practical experience using the approach; section 6 revises related work; finally, section 7 presents our conclusions and future work.

## 2 Notations of the Approach

The approach uses two notations: BPMN and Map. This section describes them briefly and justifies their selection for business process modelling and purpose analysis. Their joint use is also explained and justified. For further details about BPMN and Map, see [23] and [27], respectively.

### 2.1 BPMN

We use BPMN for business process modelling. Its creators argue that it offers a notation that is understandable by all business process users (process analysts, IS developers, process managers...). Therefore, BPMN provides a standard that fills the gap between business models and their implementation.

The notation consists of a diagram, called Business Process Diagram (BPD), whose aim is to provide a means for the development of graphical models of business process operations. A BPD is designed from a set of graphical elements that make diagrams simple to develop and easy-to-understand. The graphical elements are flow objects (activities, gateways and events), connecting objects (sequence flows, message flows, and associations), swimlanes (pools and lanes), and artefacts (data objects, annotations, and groups).

With regard to the choice of BPMN, several surveys have evaluated its adequacy for business process modelling and have compared it with other notations. These surveys are based on different criteria, such as workflow patterns (e.g. [34]), quality principles (e.g. [22]), or the BWV representation model (e.g. [28]). From the result of these studies and our own experience, BPMN has three main advantages: it is probably the most expressive notation, it is easy to use and understand, and it is receiving strong support from practitioners and vendors. As a result, BPMN is considered the de facto standard for business process modelling.

### 2.2 Map

Map is a goal-driven approach whose aim is to capture the intentions (goals) of an enterprise or system and determine the strategies that can contribute to the fulfilment of these intentions. The emphasis on the concept of strategies as ways to achieve goals distinguishes Map from other goal-based approaches. This emphasis is motivated by the fact that stakeholders do not naturally make the distinction between goals and strategies, and, as a consequence, pitfalls can arise [27]. The size of a goal model can unnecessarily increase when strategies are expressed as goals, alternative ways to make the business can be more difficult to discover, and recognizing stable elements in a business (intentions) versus more versatile ones (strategies) can be more difficult. In addition, Map promotes variability analysis at the requirements stage.

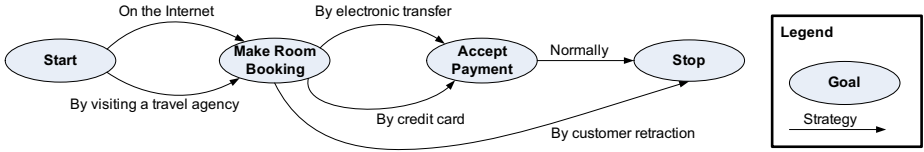


Fig. 1. Example of a map for a booking [27]

Map diagrams consist of a graph (called map) whose nodes are intentions and whose edges are strategies. An example is shown in Fig. 1. An edge entering a node identifies a strategy that can be used to achieve the intention of the node, so a map shows which intentions can be achieved by which strategies. Each map has two special intentions, Start and Stop, associated with the initial and final state, respectively. The aggregation of a source intention, a target intention, and a strategy is called section. Sections can be refined in another map.

Of all the goal-based approaches available within requirements engineering and business process reengineering, we use Map for the following reasons: it focuses on strategies to achieve goals; it has only two main concepts (goal and strategy), thus facilitating its use and understanding; and it does not deal with tasks, which we prefer to consider on a BPD instead of on a goal diagram.

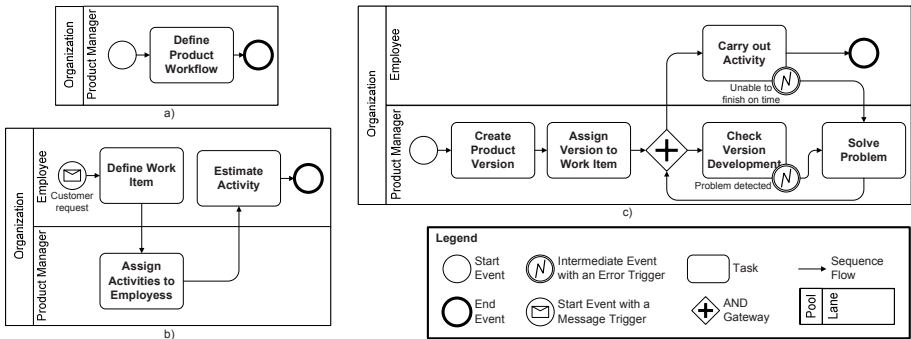
Map has been used in several projects and in different areas, including business process modelling (for detailed references, see [27]). An important advantage of Map is that it encourages customer participation and helps to solve communication problems between business people and system analysts. However, we think that other notation should be used for business process modelling. We base this opinion on the criteria that are usually analyzed when evaluating business process notations, as described above for BPMN. The use of just two main concepts (goal, strategy) is useful for purpose analysis, but it is a drawback when addressing business process modelling because the models are neither detailed nor expressive enough to have a deep understanding and knowledge of business processes.

In summary, we think that BPMN and Map can complement each other. BPMN is better suited for business process modelling, but it does not provide any mechanism for system purpose analysis so that a new business process that fulfils organizational needs is designed. Map focuses on system purpose, and the use of strategies allows its graphs to be simpler than the goal diagrams of other approaches. As a result, the approach benefits from the features of both BPMN and Map.

### 3 Case Study

As a case study, we will use the business processes for the product development of a software company (Fig. 2).

The organization develops a software product that is provided to several customers. The product is standard, so no customer has a personalized product. However, customers can request improvements in the product, and the requests are included in future versions of the product.



**Fig. 2.** BPDs for product development: a) definition of product workflow; b) request management; c) version development

The product manager defines the set of activities that has to be carried out to develop the product through product workflow. When a customer requests a new improvement, an employee defines the work item that is necessary to provide the customer with the request. Next, employees are assigned the activities that are necessary to develop the work item, and employees have to estimate how long the activities will take. The product manager is also responsible for the periodical creation of product versions, which have a strict deadline, and must decide the version in which a work item will be developed. Afterwards, employees carry out the activities in order to finish the work item and deliver the requested improvement, and the product manager checks that version development is correct. However, problems may arise while developing versions. Employees may not be able to finish the activities they are responsible for due to time constraints. If a problem arises, the product manager has to try to solve it.

### 4 Approach Description

The approach (Fig. 3) consists of three stages: organizational modelling, purpose analysis, and functional requirements specification. The first one depicts the current business environment (As-Is), which has a problem or need that could be fulfilled by an IS. The organization will change to solve the problem (To-Be), and the change will have an effect on business processes.

In the first stage, the organization for which the IS is going to be developed is modelled. For this purpose, the information gathered is a glossary, the business events, the business rules, and a role model. BPDs of the organization are created from this information. The diagrams must be validated by the end-users in order to guarantee that the organization has been properly modelled and understood. Several iterations are usually needed to get the final version.

The organizational problem or need is analyzed during purpose analysis stage. The aim is to find system strategies that can solve the organization problem, determine how to operationalize the strategies, and agree on the effect that the development of the IS may have on the business processes. This effect is the support or control that

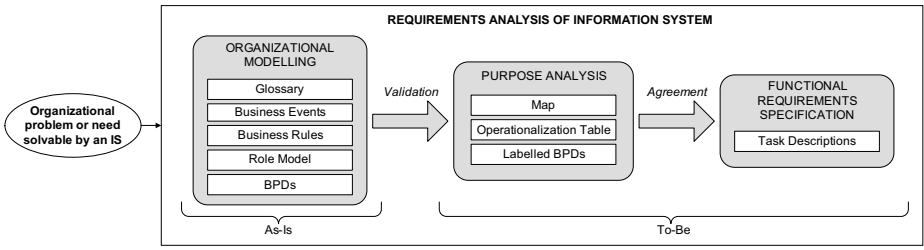


Fig. 3. Approach overview

the IS will have on them. As a result, business process elements are labelled according to the effect that the operationalization of the strategies will have on them, and changes in the business process can occur.

Finally, functional requirements are specified by means of the description of the business process tasks to be supported by the IS. Every task will have a textual template that describes it. The set of templates will be the starting point of the rest of the development stages.

For brevity, the organizational modelling stage is not described in the following sections, so only the purpose analysis and functional requirements specification stages are detailed.

### 4.1 Purpose Analysis

After organizational modelling, the system analyst has enough knowledge about the business processes to properly understand the organization activity. Nevertheless, the analyst also needs to understand the organizational problem or need. Consequently, purpose analysis is carried out from the business processes and the organizational problem or need to be fulfilled.

The introduction or modification of an IS in an organization can cause business process reengineering. An IS can initiate and supports reengineering before, during or after a process is designed [3]. In our approach, IS is a facilitator of the changes in the business processes because its effect is taken into account while designing the new processes. The new business processes (To-Be) are designed from the original ones (As-Is), the organizational needs, and the solutions that the IS can provide. IS will support business processes, and business processes are designed in terms of the IS capabilities.

Purpose analysis consists of three stages: map construction, map operationalization, and BPDs creation and labelling.

#### 4.1.1 Map Construction

The organizational problem or need is modelled in a map where the solutions that the IS can provide are analyzed. The map is created in a participative manner between the system analyst and end-users, so they can agree on the solution. First, a map is created to analyze the problem or need. Second, the goals that the end-users want to achieve in order to solve the problem through the use of the IS are modelled as intentions (nodes). Third, system features that can fulfil the end-user goals are modelled as

strategies (edges), which link the nodes. Finally, sections are refined if needed. These guidelines are based on the map construction process [27], but we have adapted them to the specific use of defining system strategies to fulfil end-user intentions.

The map that corresponds to the case study is shown in Fig. 4. The organization has been experiencing problems with delivery requests. Lack of knowledge about version development has caused requests to be delivered later than expected by customers. As a result, the customers have complained, and the strategic goal “Keep customer satisfaction” is not met. The main reason for the delay is that activity development is not always performed as planned because of the great amount of work that employees have to do. The product manager needs to be able to better project, for example, if an employee will miss working days, or if an employee has spent more time than planned on an activity. The product manager needs to foresee problems and find solutions quickly. In addition, employees need to be able to determine more accurately the time they have at their disposal to finish the activities that are in their charge of and how long these activities will take.

To solve these problems, employees wanted the IS to facilitate the work item development and to improve the knowledge they have about the status of the activities that they have to carry out. The product manager wanted the IS to improve the knowledge about the status of the versions and to minimize the time that takes a request to be delivered. The system analyst proposed system features that could fulfil these intentions and modelled them in the map in accordance with end-users.

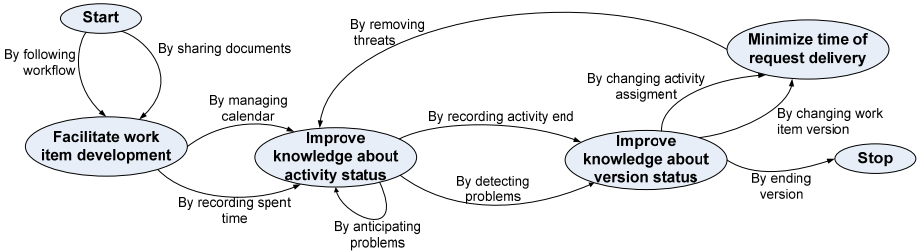


Fig. 4. Map for product development process

### 4.1.2 Map Operationalization

When the map is finished, the system analyst has to determine how to operationalize the map strategies, and come to an agreement on the effect that the operationalization will have on the old business process. Existing BPD elements can be removed or maintained, and new elements may be introduced. To facilitate this analysis, a table with three columns is created: a column to list the strategies; a column to list the BPD elements that will operationalize each map strategy and specify if the element has been removed (R), maintained (M), or it is new (N); and a column to specify the participant that will be in charge of the element.

Table 1 shows the BPD elements that operationalize each map strategy for the case study. There are several new elements: “Start activity” refers to the task in which an employee begins the performance of an activity and has to receive the necessary documents to carry out the activity; “Finish Activity” refers to the task in which an employee finishes an activity and has to share the documents related to the activity

performance; “Manage Calendar” refers to the task in which an employee divides the time that can spent in a working day; “Need to start activity” refers to the condition in which an employee must be notified that an activity has to be started in order to finish the work item before version deadline; “Change Activity Assignment” refers to the task in which a product manager changes the employee that is responsible for an activity in order to finish a work item before version deadline; “Change Work Item Version” refers to the task in which the product manager changes the version of a work item due to some problem; “Version deadline” refers to the moment in which the date of version release is reached; “Release Version” refers to the task in which the product manager releases a finished version.

**Table 1.** BPD elements that operationalize the map strategies of the case study

Map strategy	BPD element	Participant
By following workflow	Define product workflow (M)	Product Manager
	Assign Activities to Employees (M)	
	Start Activity (N)	Employee
	Carry out Activity (M)	
	Finish Activity (N)	
By sharing documents	Start Activity (N)	Employee
	Finish Activity (N)	
By managing calendar	Manage Calendar (N)	Employee
By recording spent time	Carry out Activity (M)	Employee
	Finish Activity (N)	
By anticipating problems	Estimate Activity (M)	Employee
	Need to start activity (N)	
By recording activity end	Finish Activity (N)	Employee
By detecting problems	Check Version Development (M)	Product Manager
	Problem detected (M)	
	Carry out Activity (M)	Employee
	Unable to finish on time (M)	
By changing activity assignment	Change Activity Assignment (N)	Product Manager
By changing work item version	Change Work Item Version (N)	Product Manager
By ending version	Version deadline (N)	Product Manager
	Release Version (N)	
By removing threats	Carry out Activity (M)	Employee
	Notify changes (N)	Product Manager

With regard to how the BPD elements operationalize the map strategies, we will use the map strategy “by following workflow” as example. The BPD elements that operationalize the map strategy are “Define Product Workflow” because it is the task in which the activities and documents of the product workflow are defined, and “Assign Activity to Employee”, “Start Activity”, “Finish Activity” and “Carry out Activity” because they refer to activities of the product workflow. Therefore, the workflow is followed because of the execution of these tasks.

#### 4.1.3 BPDs Creation and Labelling

Finally, the system analyst and the end-users agree upon the design of the new business process. First, changes in the BPDs are modelled, i.e., elements can be removed or introduced according to the operationalization of the map strategies. Next,



BPD elements are labelled according to the IS support on them. Tasks, events with triggers, and gateways that depict decisions are labelled as: “O” (out of the system), if the element will not be part of the IS; “IS” (controlled by the system), if the IS will be in charge of its control and execution with no human participation; or “U” (executed by a user), if the element will be executed by a person that interacts with the IS.

For the case study, Fig. 5 shows the new business processes. The business processes have changed as a result of the introduction of the IS. Several new tasks and events have been defined, and there is a new business process (calendar management) and a sub-process (problem resolution). An interesting fact is that, as acknowledged by other authors [4], the new BPDs (software development-oriented) are more detailed than the original BPDs (organizational documentation-oriented).

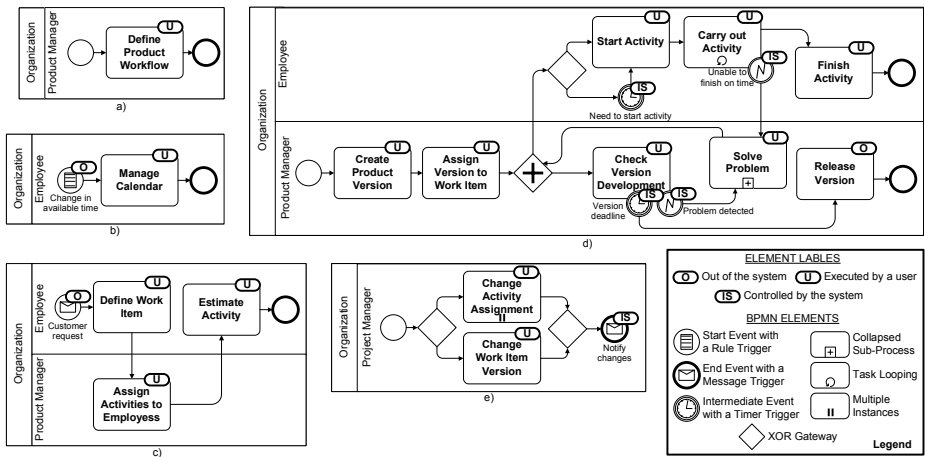


Fig. 5. Labeled BPDs for: a) definition of product workflow; b) calendar management; c) request management; d) version development; e) problem resolution

After labelling, the data objects that are input and output of every task to be supported by the system are defined. The state of the data objects before and after the execution of the task is specified. In order to keep Fig. 5 as simple as possible, the data objects are not shown for this case study.

### 4.2 Requirements Specification

In the last stage of the approach, functional requirements are specified from the labelled BPDs. For this purpose, business process tasks to be supported by the system are described through a textual template.

The content of the template is based on essential use cases [7] and task & support descriptions [19]. An essential use case is a simplified form of use case that depicts an abstract scenario for a complete and intrinsically useful interaction with a system from the perspective of users. A task & support description is a way to express what the system actors want to perform, including domain-level information and how a

new system could support an activity to solve a problem. Both essential use cases and task & support descriptions contain the fewest presuppositions about technology.

An example of task template is shown in Table 2. It corresponds to the task “Carry out Activity” of the case study. A task template includes the business process to which the task belongs, the name of the task, the role responsible for its execution, the triggers, preconditions and postconditions of the task, the input and output data and their states, a specification of the interaction between a user and the IS through user intention and system responsibility, and the business rules that affect the task.

**Table 2.** Template for the task “Carry out activity”

Business Process: Version development			
Task: Carry out Activity		Role: Employee	
Triggers			
-			
Preconditions			
-			
Postconditions			
-			
Input		Output	
Data Object	State	Data Object	State
Activity	In progress	Activity	In progress
		Time register	
User intention		System responsibility	
		1. Show the activities in progress assigned to the employee	
		2. Select an activity	
		3. Carry out activity	
		4. Indicate performance begin and end	
		5. Record time register	
		6. Update employee time	
Business Rules			
<ul style="list-style-type: none"> <li>• An employee can only carry out an activity at the same time</li> <li>• An employee can only carry out activities that have been assigned to him</li> </ul>			

All the information of a task template comes from its BPD. The name of the business process and the name of the task are the same as in the BPD. The role is the participant in the business process that is in charge of the task. The triggers correspond to the events with a trigger that precede the task in the business process and are part of the IS. The preconditions correspond to the gateways that precede the task, represent decisions, and are part of the IS. The postconditions correspond either to the gateways that follow the task, represent decisions that can make the business process iterate and are part of the IS, or to the end events with a trigger that follow the task and are part of the IS. The input and output of the task are its data objects in the business process. User intention and system responsibility are defined from the behaviour of the participant in charge of the task when executing it and how the system will support it. User intention may include actions that do not represent interactions with the system, and the system responsibility must be agreed upon with the end-user. Finally, the business rules specified in the template correspond to business rules that define or constrain the task, could not be modelled graphically, and have to be supported by the system.

After every business process task has been described, the subsequent development stages will be based on the task templates in order to provide the organization with an IS that fits its needs, its structure, and its behaviour.

## 5 Practical Experience

As explained above, the approach is the result of a project with a company, CARE Technologies. The purpose of the project is to solve problems related to the requirements stage by trying to link business and software domains.

After analyzing the requirements practices of the company, we identified problems related to business understanding, purpose analysis, and communication with end-users. The company uses OO-Method [24], a methodology for automatic software generation based on conceptual modelling. The data conceptual schemas consist mainly of a class diagram that is enriched with functional information about the result of class service execution. Analysts just provide some textual description about the requirements and validate them on the class diagram or on the generated application.

Although analysts feel comfortable with this technique, we think it could be improved. Some authors have stated that class diagrams alone might not be appropriate for communicating and verifying requirements, that there are few studies addressing the ability of end-users to understand class models, and that they can be complex for people that have not been trained in object-oriented modelling [12]. In addition, objects might not be a good way of thinking about a problem domain [33].

The approach has been used in four real projects in order to evaluate it and try to find improvements. It has been refined gradually based on the comments of both customers and analysts. The organizations for which the systems were developed were an apartment rental company, a car rental company, the organization of a golf tournament, and the organization of this case study. They were small/medium size projects. CARE had developed software for the organizations previously, so both the technique they usually use and our approach could be compared.

The system purpose was analyzed on the BPDs before the case study project. The reason for this is that the purpose analysis of the other projects was less complex, and thus easy to analyze. The introduction of the new IS did not change the business processes significantly, and the result was automation rather than reengineering. However, the system analysts said that some technique for purpose analysis would be helpful for the product development project.

We held between 2 and 5 meetings with end-users to obtain the business process models of the organizations, 1 meeting for purpose analysis, 1 or 2 meetings to define the task templates, and 1 meeting to validate the entire requirements specification. Another meeting was held in order to talk about the experience with end-users and analysts. Each meeting took approximately 2 hours.

As expected, the end-users stated that they could understand and validate the requirements models of the approach more easily than the class diagrams, thus facilitating communication and interaction. They also felt more involved in system development and claimed that they had a more participative attitude.

When asking analysts about the usefulness of the approach, we obtained different opinions. Although all the analysts stated that the approach allowed them to better

understand the organizations, the system purpose, and, consequently, the requirements, there were some analysts who did not think that the approach could improve their job significantly and would probably not use it.

We do not find these comments about the approach discouraging. The analysts who did not think that the approach was very useful were senior analysts that are already very skilled in using OO-Method and interacting with customers. They usually model the systems while the customer describes what the system should do, so they can quickly generate it, validate it, and fix it if needed. However, most of the junior analysts, who have less experience in dealing with customers and, therefore, in understanding what is needed, considered that the approach could really help them.

We think these results are a reflection of common practices in IS development. Models are only used when they are believed to be useful [10]. In our case, some senior analysts do not think that the approach can accelerate their job, whereas junior analysts think that it can improve their performance.

Another interesting subject that arose while discussing the approach was the viewpoint to use when defining strategies in the map. We recommended the system analysts to include both system and enterprise strategies in order to better analyze the problem. However, they argued that the company mission was software development, not business consultancy. System analysts believed that BPDs and end-users goals were enough to properly understand the organization. Therefore, they did not consider business strategies that would not be supported by the IS to be useful for their work.

Finally, this practical experience has some limitations. First, the approach has to be used in more projects to draw definite conclusions. Second, the opinions of end-users and the analysts were obtained by discussing with them informally, so we are now designing a form to survey the next projects. Last but not least, we want to assess the approach by means of experiments with students and analysts from other companies.

## 6 Related Work

The need of organizational modelling and system goal analysis has been widely acknowledged within requirements engineering. Many approaches consider them to be the first step in software development, and some approaches use business process modelling. Nevertheless, the approaches usually focus on only one of the issues, and the use of models and techniques that facilitate the communication is not common.

Goal-oriented approaches have played an important role within requirements engineering. They try to solve the problem of systems that do not properly respond to organizational and user needs. Despite the acknowledged contributions of goal-oriented approaches, they have some weaknesses [26]. In our opinion, the main weaknesses are that goals might not provide a good starting point for requirements analysis, goal reduction is not straightforward, and most of the approaches do not pay enough attention to business concerns and business process reengineering. Apart from Map, two well-known approaches are *i\** [35] and KAOS [9].

*i\** is focused on the modelling of dependencies among the organizational actors in order to achieve organizational goals. It has been used in several software development methods, such as Tropos [6] or RESCUE [21]. However, several

weaknesses have been identified by practitioners (e.g. [15]). *i\** diagrams might be too complex, and they should better support granularity and refinement.

KAOS requirements models are built from organizational goals. These goals are systematically refined to operational requirements through refinements patterns. KAOS also uses an object model, an agent responsibility model, and an operation model. When comparing the use of KAOS with Map, we have obtained simpler goal models with Map because of its focus on strategies. In addition, KAOS does not provide any mechanism to model and analyze business processes.

Among organizational modelling-based approaches, EKD [5] and ARIS [11] provide ways of analyzing an enterprise by using enterprise modelling. EKD is composed of a goal model, a business rules model, a concepts model, a business process model, an actor and resources model, and a technical components and requirements model. ARIS consists of five views: organizational view, data view, control view, function view, and product/service view. In our opinion, when using EKD and ARIS for tailored software development, their requirements specifications should be more detailed, and they should be more focused on goal analysis. EKD also lacks tool support that facilitates the development and maintenance of all its models. In addition, as we have stated above for Map, BPMN is better suited for business process modelling than the notations that EKD and ARIS propose.

Some approaches use UML for organizational and business process modelling (e.g. [14]). These approaches use elements that are close to those elements used in the software development area. This fact is a drawback, because the models are easy to use and understand by system analysts but might be too complex to be validated by end-users. In addition, the UML-based approaches do not focus on goal analysis.

## 7 Conclusions and Future Work

Requirements analysis is still a stage of software development where mistakes are common. Therefore, it can be the source of problems in subsequent development stages and can cause an IS not to fulfil the real needs of the organization where the IS has to be modified or introduced. Some of the mistakes detected in practice are the lack of understanding of the business by system analysts, the lack of focus on system purpose, and miscommunication between business people and system analysts.

This paper has described an approach to try to prevent these problems based on the modelling of an organization by means of BPMN and the goal/strategy Map approach. The approach allows system analysts to properly understand and analyze the organization, its needs, and the system goals in a participative way with end-users. Business people and system analysts share a common language that is understandable to both of them thanks to BPMN, Map, and task templates. BPDs are the basis for the end-user to validate that the organization structure and behaviour have been properly understood so that the system analyst can propose solutions based on the system purpose. Furthermore, the approach tries to mitigate the weaknesses of a separate use of BPMN and Map, and benefit from the advantages of their joint use.

Apart from the improvement in the surveys, the next steps in the project are the development of a tool that supports the approach, the introduction of a technique for the analysis of non-functional requirements, and the linking of the approach with

OO-Method. In addition, we want to extend the approach by introducing information about the user interface in the task template in order to derive an abstract description of the interaction between the users and the IS.

## References

1. Agarwal, R., Prabuddha, D., Sinha, A.P.: Comprehending Object and Process Models: An Empirical Study. *IEEE Transactions on Software Engineering* 25(4), 541–556 (1999)
2. Alexander, I., Bider, I., Regev, G.: REBPS 2003: Motivation, Objectives and Overview. Message from the Workshop Organizers. In: *CAiSE Workshops* (2003)
3. Attaran, M.: Exploring the relationship between information technology and business process reengineering. *Information & Management* 41, 585–596 (2003)
4. Becker, J., Kugeler, M., Rosemann, M. (eds.): *Process Management*. Springer, Heidelberg (2003)
5. Bubenko, J., Persson, A., Stirna, J.: *EKD User Guide* (2001), <http://www.dsv.su.se/~js>
6. Castro, J., Kolp, M., Mylopoulos, J.: Towards requirements-driven information systems engineering: the Tropos Project. *Information Systems* 27, 365–389 (2002)
7. Constantine, L., Lockwood, L.: *Software for Use*. Addison-Wesley, Reading (2002)
8. Curtis, B., Kellner, M., Over, J.: *Process Modelling*. *Communications of the ACM* 35(9), 75–90 (1992)
9. Dardenne, A., Lamsweerde, A., van Fickas, S.: Goal-directed Requirements Acquisition. *Science of Computer Programming* 20, 3–50 (1993)
10. Davis, I., et al.: How do practitioners use conceptual modelling in practice? *Data & Knowledge Engineering* 58, 359–380 (2006)
11. Davis, R., Brabänder, E.: *ARIS Design Platform*. Springer, Heidelberg (2007)
12. Dobing, B., Parsons, J.: Understanding the role of use cases in UML: a review and research agenda. *Journal of Database Management* 11(4), 28–36 (2000)
13. Dumas, M., van der Aalst, W., ter Hofstede, A. (eds.): *Process-Aware Information Systems*. Wiley, Chichester (2005)
14. Eriksson, H., Penker, M.: *Business Modelling with UML*. John Wiley and Sons, Chichester (2000)
15. Estrada, H., Rebollar, A.M., Pastor, Ó., Mylopoulos, J.: An Empirical Evaluation of the i\* Framework in a Model-Based Software Generation Environment. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2006*. LNCS, vol. 4001, pp. 513–527. Springer, Heidelberg (2006)
16. IIBA. *Business Analysis Body of Knowledge* (2006), <http://www.iiba.com>
17. Holtzblatt, K., Beyer, H.: Requirements gathering: the human factor. *Communications of the ACM* 38(5), 31–32 (1995)
18. Kirikova, M., Bubenko, J.: Enterprise Modelling: Improving the Quality of Requirements Specification. *Information Systems Research Seminar in Scandinavia, IRIS-17* (1994)
19. Lauesen, S.: Task Descriptions as Functional Requirements. *IEEE Software* 20(2), 58–65 (2003)
20. Luftman, J., Raymond, R., Brier, T.: Enablers and Inhibitors of Business-IT Alignment. *Communications of AIS* 1(11), 1–33 (1999)
21. Maiden, N., Jones, S.: *An Integrated User-Centered Requirements Engineering Process, Version 4.1* (2004), <http://hcid.soi.city.ac.uk/research/Rescue.html>
22. Nysetvold, A., Krogstie, J.: Assessing Business Process Modelling Languages Using a Generic Quality Framework. In: *EMMSAD 2005, CAiSE Workshops* (2005)

23. OMG: Business Process Modelling Notation (BPMN) Specification (online) (2006), <http://www.bpmn.org>
24. Pastor, O., Molina, J.C.: *Model-Driven Architecture in Practice*. Springer, Heidelberg (2007)
25. Reich, B., Benbasat, I.: Factors That Influence the Social Dimension of Alignment Between Business and Information Technology. *MIS Quarterly* 24(1), 81–113 (2000)
26. Rolland, C., Salinesi, C.: Modeling Goals and Reasoning with Them. In: *Engineering and Managing Software Requirements*, pp. 189–217. Springer, Heidelberg (2005)
27. Rolland, C.: Capturing System Intentionality with Maps. In: *Conceptual Modelling in Information Systems Engineering*, pp. 141–158. Springer, Heidelberg (2007)
28. Rosemann, M., et al.: A Study of the Evolution of the Representational Capabilities of Process Modeling Grammars. In: Dubois, E., Pohl, K. (eds.) *CAiSE 2006*. LNCS, vol. 4001, pp. 447–461. Springer, Heidelberg (2006)
29. Sommerville, I., Sawyer, P.: *Requirements Engineering: A Good Practice Guide*. John Wiley and Sons, Chichester (1997)
30. Stirna, J., Persson, A., Sandkuhl, K.: Participative Enterprise Modeling: Experiences and Recommendations. In: Krogstie, J., Opdahl, A., Sindre, G. (eds.) *CAiSE 2007*. LNCS, vol. 4495, pp. 546–560. Springer, Heidelberg (2007)
31. The Standish Group. *Chaos Reports*, <http://www.standishgroup.com>
32. Taylor-Cummings, A.: Bridging the user-IS gap: a study of major information systems projects. *Journal of Information Technology* 13, 29–54 (1998)
33. Vessey, I., Coner, S.: Requirements Specification: Learning Object, Process, and Data Methodologies. *Communications of the ACM* 37(5), 102–113 (1994)
34. Wohed, P., et al.: On the Suitability of BPMN for Business Process Modelling. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) *BPM 2006*. LNCS, vol. 4102, pp. 161–176. Springer, Heidelberg (2006)
35. Yu, E.: *Modelling Strategic Relationships for Process Reengineering*. PhD Thesis, University of Toronto (1995)
36. Zave, P., Jackson, M.: Four Dark Corners of Requirements Engineering. *ACM Transactions on Software Engineering and Methodology* 6(1), 1–30 (1997)