A Business-Oriented Approach to Requirements Elicitation

Adam Przybyłek
Faculty of Mathematics, Informatics and Mechanics, University of Warsaw, Warszawa, Poland

Keywords: Requirements, Business Process Modeling, BPM, Use Cases, Traceability, UCDA, UML, Activity Diagram.

Abstract: "The hardest single part of building a software system is deciding precisely what to build" (Brooks, 1987). Faulty requirements analysis is a major reason for project failure or unsatisfactory information systems that do not fulfill business needs. Although it has been long recognized that system requirements can only be understood within the context of the business processes and business modeling has become the initial phase of most software processes, the transition between business modeling and requirements gathering is still a challenge for research. Moreover, existing work in this area tends to introduce accidental difficulties. This paper reports the results of an action research conducted for elaborating a Business-Oriented approach to Requirements Elicitation. Our approach integrates Requirements Engineering with Business Process Engineering and derives system requirements based on business process models. This ensures that system requirements meet real business needs. The proposed approach is illustrated by a real-world example.

1 INTRODUCTION

Information systems (ISs) sustain the underlying business processes in most of today's organizations. However, few ISs are developed with explicit consideration of the business processes they are supposed to support. A recent survey (Wolf and Harmon, 2012) shows that in 179 out of 372 organizations that were examined, the support provided by the existing software systems was never or occasionally consistent with the business processes. The sources of this misalignment can be traced to the "essential" and "accidental" difficulties of the requirements engineering process. We organized these difficulties in Table 1. For each essential difficulty we identified the corresponding accidental difficulty that compounds it. We also pointed out the challenges that, if met, would reduce the accidental problems. The essential difficulty is inherent in what one is trying to accomplish; it is part of the problem itself. In turn, the accidental difficulty is introduced through inadequate practices; it comes from the way one is tackling a problem (Brooks, 1987). Thus, analysts should focus their attention on solutions for the essential difficulties, and apply techniques and methods which help them avoid the accidental difficulties.

We explain how to read Table 1 by taking the first row as an example. Understanding what the customer needs is an essential difficulty, because customers do not really know what they need. System analysts have to make the customer realize its needs. Thus, this difficulty is inherent in the problem of requirements elicitation and we cannot avoid it. However, inadequate requirements elicitation practices unnecessarily exacerbates this difficulty. For instance, customers are not involved enough in the requirements engineering process or system analysts just model what the customer is saying, while customers usually can not articulate requirements that truly reflect their business needs. The challenge is to build relevant business process models and then to derive system requirements from these models.

To address the challenges presented in Table 1, we combine techniques from both the fields of Business Process Engineering and Requirements Engineering and define BORE: a Business-Oriented approach to Requirements Elicitation. Our approach allows for deriving system requirements from business process models and enables traceability between business processes and the corresponding system requirements. This ensures that system requirements meet real business needs and that there are no superfluous requirements.

The rest of the paper is organized as follows. Next section reports the design of our research. Section 3 presents the general description of BORE, while its succeeding subsections describe the stages of the approach in detail. Section 4 demonstrates

Przybyłek A..
A Business-Oriented Approach to Requirements Elicitation.
DOI: 10.5220/0004887701520163
In *Proceedings of the 9th International Conference on Evaluation of Novel Approaches to Software Engineering* (ENASE-2014), pages 152-163
ISBN: 978-989-758-030-7

Table 1: Essential and accidental difficulties of the requirements engineering process.

Essential difficulties	Accidental difficulties	Challenges
understanding what the customer needs <= customers do not really know what they need (Faulk, 1997) or have only a vague picture of their requirements (Maciaszek, 2005)	inadequate requirements elicitation practices <= customers are not involved enough in the requirements engineering process; <= customers usually can not articulate requirements that truly reflect their business needs	business analysis must precede requirements elicitation in order to obtain a deep enough understanding about the organization; business process models must drive requirements elicitation
effective communication among stakeholders due to the "culture gap" (Taylor-Cummings, 1998) between business and system domains; <= a requirements specification has many audiences with different viewpoints and knowledge (Faulk, 1997)	rework and miscommunications due to many notations <= business processes are not consistently documented and different stakeholders tend to use different notations, conventions and techniques to represent them (Monsalve et al., 2010); <= business models are usually done using different notation than that of software models; a recent survey (Harmon & Wolf, 2011) shows that the vast majority of business people (72%) use BPMN, while UML, which is a standard in software design, was listed only by 18% respondents (the respondents could indicate more than one notation; 8% chose EPC).	if we could use the same notation through the whole project all stakeholders can share the same work products
frequent and arbitrary changes to requirements (Faulk, 1997) <= business processes change due to technology advancements, pressure from competitors, new legislation, etc.; in this case, the requirements may need adaptations to align the evolving business processes	deficiencies in backward traceability (Ravichandar et al., 2007; Gotel & Finkelstein, 1994) make it difficult to keep consistency between documentation and the underlying system <= business processes are not linked to the system requirements and thus evolve independent from the IS;	the documentation must link business processes to artefacts of an analysis, design and implementation in an explicit and traceable manner;

how to use BORE. Section 5 discusses related work. Finally, Section 6 concludes with a review of the contribution.

2 RESEARCH DESIGN

This research reports on the action research project at a private university. The university was experiencing problems with an overwhelming amount of work during the admission and timetabling period. It was usual for administrative staff to work overtime, evenings or weekends. Moreover, the highly competitive market of educational services was forcing the university to improve the teaching process and communication among lecturers and students by leveraging IT solutions. Nevertheless, the authorities did not have a clear and detailed idea about the real needs and were not completely sure what they wanted.

The action research methodology aims at providing value to the participant organization but acquiring new knowledge at the same time (Sjoberg et al., 2007). The research objective was to develop a new approach that (1) supports requirements elicitation under conditions of uncertainty about client needs; and (2) makes that the system requirements are in alignment with and provide support for the underlying business processes. The practical objective was to analyze the business situation, identify opportunities for improvements, and specify the requirements of an IS that supports the business needs.

The participant organization was selected for this research due to: (1) its willingness to undergo a process of innovation; (2) its interest for the project that was founded from an external grant (it was free for the organization); (3) its accessibility, and (4) the application domain that was known by the researcher.

Four main approaches to data collection were used in this research. These were: semi-structured and unstructured interviews, apprenticing, workshops, and document analysis.

The researcher met with seven customer representatives at every level in the organization. These representatives were selected by the research coordinator, who was an employee of the university and knew the staff. Then, the representatives were trained in business process diagrams, activity diagrams and use case diagrams. As part of their training, they had to model business processes and draw a use case diagram based on a given case study.

The customer stakeholders participated in the project to varying degrees. During As-Is business process modeling, the researcher created models and then the customer stakeholders were consulted to validate the models. This approach is called consultative participation. In contrast during business process engineering and use-case modeling, the customer stakeholders collaboratively developed models in facilitated group sessions. This type of participation is consensus-driven in the sense that it is the stakeholders who "own" the model and hence decide its contents (Bubenko et al., 2001), while the main role of the researcher is to facilitate the discussion, make suggestions, and gain feedbacks.

Various notations have been proposed to model business processes, e.g.: BPMN (Business Process Model and Notation), UML, Petri Nets, DFD (Data Flow Diagram), IDEF3 (Integrated DEFinition Method 3), RAD (Role Activity Diagrams) and EPC (Event Driven Process Chain). Today, the most prominent among these are BPMN and UML Activity Diagram. BPMN is more complex but at the same time more powerful (BPMN has more than 100 graphical elements). Nevertheless, Marcinkowski (2010) found that activity diagrams provide adequate modeling support for the purpose of business analysis.

Since our approach promises seamless transition from business process models to use case models, it needs the same notation for both domains. Thus, our natural choice was to use UML Activity Diagram and its extension called Business Process Diagram (if the reader is not familiar with these diagrams, we recommend reading Eriksson and Penker (2000) and Wrycza et al., (2005)).

Business Process Diagram introduces a set of stereotypes that illustrates the interaction between the processes, the resources that participate in the processes and the people acting in the processes. It describes how work is done within the business environment. The core business modelling element is represented by stereotyping an activity to a «process» or by a special icon. A process can span over multiple swimlanes to represent the fact that multiple roles can cooperate to its execution. Resource instances are represented as objects. Input objects are usually placed on the left of the process and output objects are placed on the right. A result produced by one process can constitute input to another process. The people resource is placed below or above the process and is linked to it by a dependency stereotyped by «control». For more details about Business Process Diagram, the reader is referred to (Eriksson and Penker, 2000).

3 APPROACH DESCRIPTION

BORE is based on the following observations: (1) Business Process Improvement (BPI) and IT planning are closely related; (2) customer stakeholders are often not completely sure about their real requirements and are not aware of what support they can expect from an IS; and (3) business analysis and requirements elicitation consist of overlapping activities and bear much resemblance to each other. These observations give rise to the corresponding assumptions: (1) BPI should be viewed as preliminary to and integrated with the development of IS; (2) understanding the business processes is a precondition for adequate requirements elicitation; and (3) business process models can be used for requirements elicitation.

Wherever possible, we attempted to leverage existing, well-established techniques, methods and notations in business analysis and requirements engineering rather than developing new ones. Particularly, we borrowed such means as Joint Application Design, interviewing, apprenticing, meta-plan, use-cases, and Eriksson-Penker Business Extensions.

BORE is a three-stage approach (see Figure 1). These stages are presented in subsequent subsections.

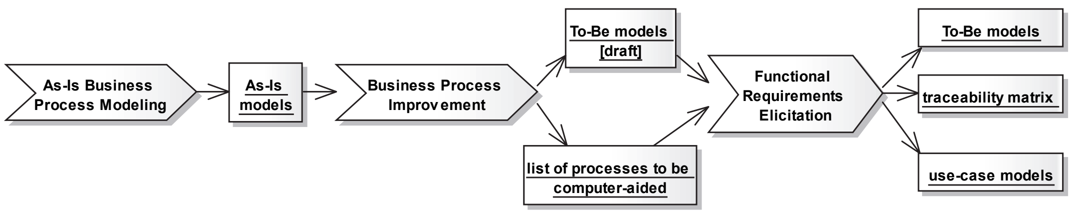

Figure 1: Overview of BORE.

3.1 As-Is Business Process Modeling

The first stage of our approach aims at understanding the organization for which an IS is going to be developed. Business process models are the main deliverables of this stage. Several techniques are available to transfer knowledge from customer stakeholders to analysts. At this stage, we recommend interviewing, apprenticing and looking through existing organizational documents.

Interviews can be guided by focus questions such as: What are the main processes of the organization?; How are these processes related?; What activities do these processes consist of?; Which information and material flows do these processes consume and produce?; Who is responsible for performing and supporting these processes? (Bubenko et al., 2001). A complementary way to obtain this information is through mining event logs (if ones exist) generated by legacy systems (Przybyłek, 2013a; 2013b; 2013c).

The gathered information is used to outline an overall picture of the business on process diagram (Eriksson and Penker, 2000). The diagram must include all fundamental enterprise-wide processes. Each complex process contains a number of activities that are performed as part of the process. Each of these activities can be considered a process of its own and as a sub-process to the containing process. Thus, we can progressively apply functional decomposition and model the internal structure of each complex process at a lower level activity diagram (Przybyłek, 2007). A process can be broken down into smaller sub-processes as long as an elementary level is achieved. A process is elementary if it is performed in one location at one time and leaves the business in a consistent state (Frost and Allen, 1996). The obtained diagrams should be validated by customer stakeholders. Usually, several iterations are needed to get the final versions.

Probably the most efficient technique for understanding the details of business processes and their connections is apprenticing. In this technique, the analyst is the apprentice, with the worker as the master craftsman. The analyst sits alongside the worker at the normal workplace and receives a running commentary on the work as it happens. Almost everyone is good at explaining what they are doing while they are doing it. As the work is observed and explained, the analyst sketches a model of each task and its connections with the other tasks. As the models are built, the analyst feeds them back to the worker to obtain confirmation that they are correct and to raise questions about any areas of uncertainty (Robertson, 2013).

The resulted models (As-Is) describe how the business is working today at any level of abstraction from enterprise-wide processes to single tasks performed by single people. For various guidelines that can support business process modeling, the reader is referred to (Vara, 2011).

3.2 Business Process Improvement

Once the business processes have been modeled, we have to agree what part of the business is to be automated. Two important factors that must be considered are costs and benefits. Note, that automating processes for the sake of automation does not lead to significant improvements (Weerakkody and Currie, 2003). Thus, as suggested by Hammer (1990), instead of blindly automating manual processes, the processes are reengineered while taking advantages of the possibilities for automation.

At this stage a workshop with key stakeholders should be held. The role of the workshop is to bring stakeholders together for a common purpose, for a short, intensive period. The role of customer stakeholders is to give feedback about the findings and suggestions presented by the analysts, make their own proposals, and provide more knowledge about the current situation (Lehtola et al., 2009). The most crucial technique at the workshop is brainstorming, which is one way of inventing. Brainstorming contains two phases: the generation phase, where ideas are collected; and the evaluation phase, where the collected ideas are discussed (Leffingwell and Widrig, 1999). To foster a creative atmosphere, all ideas are acceptable, and no one can slow the process down by criticizing or debating ideas. The aim is to be as imaginative as possible, and to get input from all stakeholders, often by using the ideas of others to trigger a different idea of their own (Robertson, 2013). Some of the ideas may seem silly, but they must still be produced to help come up with the good ideas. An idea may be impracticable as is, but a mutation of it need not be. When the idea-generation phase terminates, it is time to initiate idea reduction (Leffingwell and Widrig, 1999). The facilitator walks through each idea and asks the submitter to provide an explanation. Then, the group discusses the ideas and eliminates those that are not worthy of further consideration.

The first step at the workshop is to identify improvement opportunities in the business processes. By analyzing the existing processes, the participants look for redundancies and bottlenecks. This step should be supported by dedicated simulation software (Gawin and Marcinkowski, 2013). Next, the participants identify and discuss new organizational needs that could be fulfilled by IT. They also analyze how IT can reshape the way business is done today in order to create an environment in which the staff exploits information more effectively. Since new business processes may occur after this stage, the process models generated at the previous stage must be adjusted.

Once elementary business processes (EBPs) are revised, they must undergo a prioritization process to decide which of them and in what extent we are going to computerize. For each EBP there are three possibilities. A process may be either performed automatically by an IS, or supported by an IS, or performed manually. The abbreviations “A”, “S”, and “M” in the process descriptions represent these situations. A process is automated when the IS performs it without human participation. A process is supported when a user interacts with the IS to perform the process.

A survey of prioritization techniques can be found in Bendakir et al., (2013). In our approach, for each EBP customer stakeholders consider how an IS may help a worker complete the process and assess the potential benefit using a Low-Medium-High scale. In turn, supplier stakeholders estimate the cost of computerization. In many cases, one EBP can be computer-aided more readily as a result of having computerized another EBP. If computerization of a certain EPB is irrational without computerization of other EPBs the cost is estimated with an assumption that the closely dependent EPBs are also computerized. All closely dependent EPBs are explicitly listed for each process.

3.3 Functional Requirements Elicitation

At this stage, workshop is still the main technique for gathering information and collaborative decision-making. The business processes that have been designated to computerization, are used to derive functional requirements that the IS should supply (and that the business needs). When reviewing literature, several concrete procedures on how to transform business process models into use case models can be found (Dijkman and Joosten, 2002); (Štolfa and Vondrák, 2004); (Vara et al., 2008). However, from our experience, it has seemed to us that the transformation process cannot be algorithmized, and so we provide only general guidelines that support this stage. First of all, we must look from both the perspective of the system (asking what will make up a well-defined use case?) and from the perspective of the business process (asking what is needed from the IS?) (Eriksson and Penker, 2000). A well-defined use case must specify a complete functionality which yields an observable result of value for one or more actors.

We suggest to start the transformation from the processes labeled with ‘S’. Usually, we must create a use case for each such process. Then, the partition at which this process is occurring is mapped to an actor associated with the use case. Next, we move to the processes labeled with ‘A’. Usually, they are computerized as a part of the other use cases that have been already created. If we mapped an entirely automated process to its own use case, the use case would not be associated with any actor. According to UML, this situation is not valid.

Note, that an organization may change due to the IS deployment, and the change may have an effect on the business processes. Usually, some EBPs disappear, and the new ones that reflect new ways of running the business are introduced. Thus, we must once again adjust the process models. The new models (To-Be) are designed from the original ones (As-Is) and the support that the IS provides (Vara et al., 2008).

It is important to ensure that the IS will solve the real business needs. Therefore each use case proposed for the system must have its origin in at least one business process (a use case is said to be traced from that/this process). Note that, it is possible to have several use cases coming from the same business process as well as a single use case supporting more than one process. Among numerous techniques for tracing use cases (summarized in (Gotel and Finkelstein, 1994) and (Torkar et al., 2012)), we recommend traceability matrices. A traceability matrix is typically implemented as a table or a spreadsheet. The processes are associated to the rows and the use cases are associated to the columns of a matrix. When a process is related to a use case, a mark is placed in the intersecting cell.

When the use case diagram and the traceability matrix are accomplished, the next step is to document every identified use case in detail. The additional information is discovered by interviewing potential users of the system. Then, all use cases are analyzed to solve conflicts and inconsistencies. Here techniques and approaches from traditional requirements analysis (see (Maciaszek, 2005)) may be applied. The resulted use case model specifies a software system that adequately supports the business processes of the organization and is the starting point for the rest of the development phases.

4 BORE IN ACTION

This section reports a project conducted in a private university in Poland. The university staff provided a set of narratives of the current business processes. The employee who was most knowledgeable about the issue of interest was interviewed. For the sake of space and brevity, we omit some of the processes, simplify others, and present only a few artifacts developed during the project. Figure 2 gives a top level overview of the main business processes, while a brief description of each process is reported below.

Admission Process (Figure 3)**.** As a formal prerequisite to be eligible for admission to Bachelor's program, an applicant need to have a matriculation certificate. Admission to Master's program requires the satisfactory completion of an undergraduate (bachelor's) degree. All candidates must submit an application form and a copy of the relevant diploma. They must also pay a non-refundable recruitment fee. Admission Committee ensure each application is complete. If the number of applicants exceeds the number of available places, the applicants undergo a process of evaluation. Candidates for undergraduate level are evaluated on the basis of their matriculation grades. Master's program require candidates to pass an entrance examination. The exam is prepared, conducted and reviewed by Examination Board. Next, Admission Committee draws up a list of the strongest candidates who are initially accepted. Before the final list of the admitted students is announced, candidates must pay all tuition fees for the academic semester and send their original diplomas.

Analysis of the Business Environment. The chancellor and heads of departments continuously track changes in job offers, to adjust the curriculum to the needs of the market. IT magazines, job portals and websites of other universities are main sources of the knowledge. The authorities also have to take into account program basics defined by the government.

Marketing. The approved curriculum must be presented to a wide range of potential students. Thus, the PR staff create marketing materials that are disseminated through press, radio and TV. They also elaborate a content of the university's website.

Evaluation of Lecturers. After each semester an assistant to the dean ranks lecturers on the basis of surveys and visitations. The surveys are conducted among students by the assistant, while the visitations are carried out by heads of departments.

Assigning Courses to Lecturers. Heads of departments have to map courses to lecturers. First, courses are offered to those lecturers who were employed in the previous academic year and who were highly rated by students. If a new lecturer is employed, his/her research background is evaluated and the payment is negotiated individually. The final decision about the employment is taken by the chancellor. The assignment of a lecturer to courses is set out in a document known as a workload card, which must be signed by the chancellor.

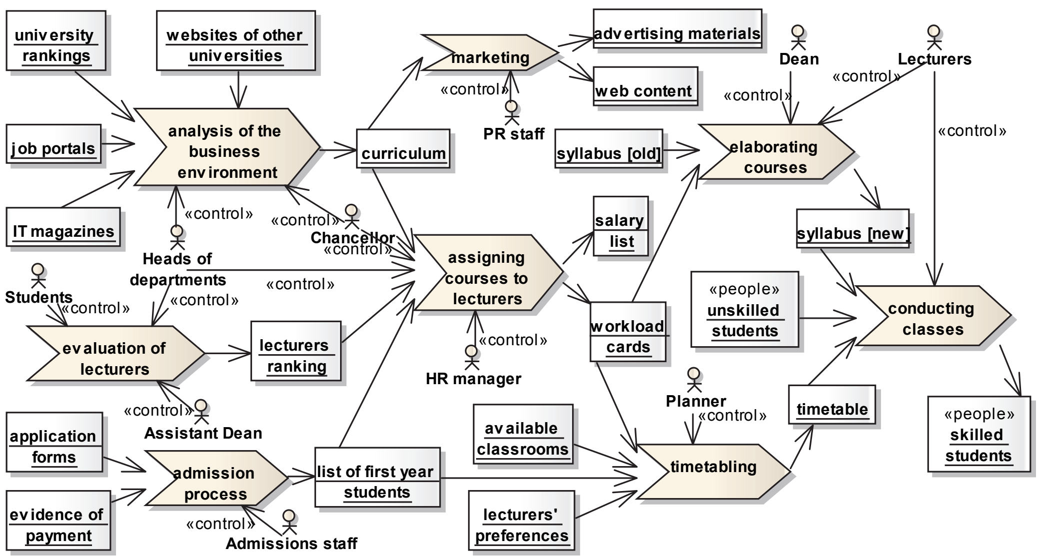

Figure 2: Top level As-Is business process model.

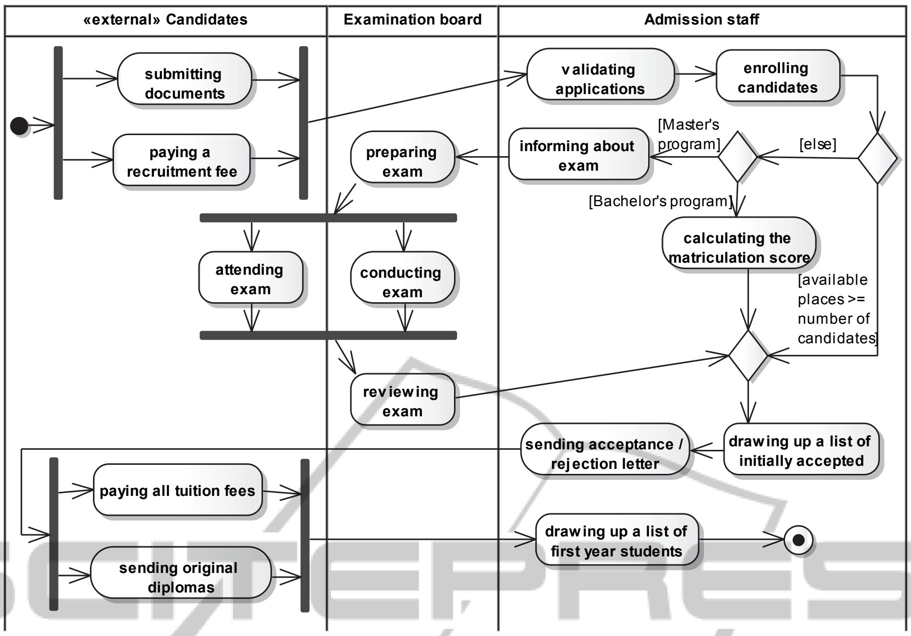

Figure 3: As-Is business model for the admission process.

Table 2: Computerization decision for the admission process.

id	elementary business process	benefits from computerization	cost to implement	dependent processes	decision
1	submitting documents	high	low	4	S
2	paying a recruitment fee	low	high		M
3	validating applications	moderate	high	1	S
4	enrolling candidates	moderate	low		A
5	informing about exam	low	low	3	S
6	preparing exam	low	moderate		S
7	conducting exam	high	low	6	S
8	reviewing exam	high	low	6, 7	A
9	calculating the matriculation score	high	low	3	A
10	drawing up a list of initially accepted	low	low	3, 8, 9	S
11	sending acceptance / rejection letter	low	low	10	A
12	paying all tuition fees	low	high		M
13	sending original diplomas	low	high		M
14	drawing up a list of first year students	low	low	10	S

Timetabling. On the basis of the workload cards, lecturers' preferences and classroom availability, the timetable is drawn up by a planner.

Elaborating Courses. Once courses are assigned, lecturers have to prepare corresponding syllabuses and modules. A lecturer can follow the syllabus that was used during the previous year or custom it to his/her own vision. All syllabuses must be approved by the dean. Detailed course curricula, rules of obtaining credits, and lists of compulsory literature are displayed in the showcases.

Conducting Classes. Lecturer familiarize students with details of curricula, rules of obtaining credits, and consultation hours. Lecturer conduct classes according to the timetable.

During the workshop, stakeholders decided to computerize the following processes: admission, assigning courses to lecturers, timetabling, and evaluation of lecturers.

Every single business process was reviewed to consider the potential support from IT. The results for the admission process are summarized in Table 2.

Figure 4 shows the To-Be business model for the admission process, which describes the way in which the university wants to operate after deploying the IS. Note, that the anticipated IS is represented by a new partition, and that the university requires candidates to apply online. Instead of labeling each process with "A", "S", or

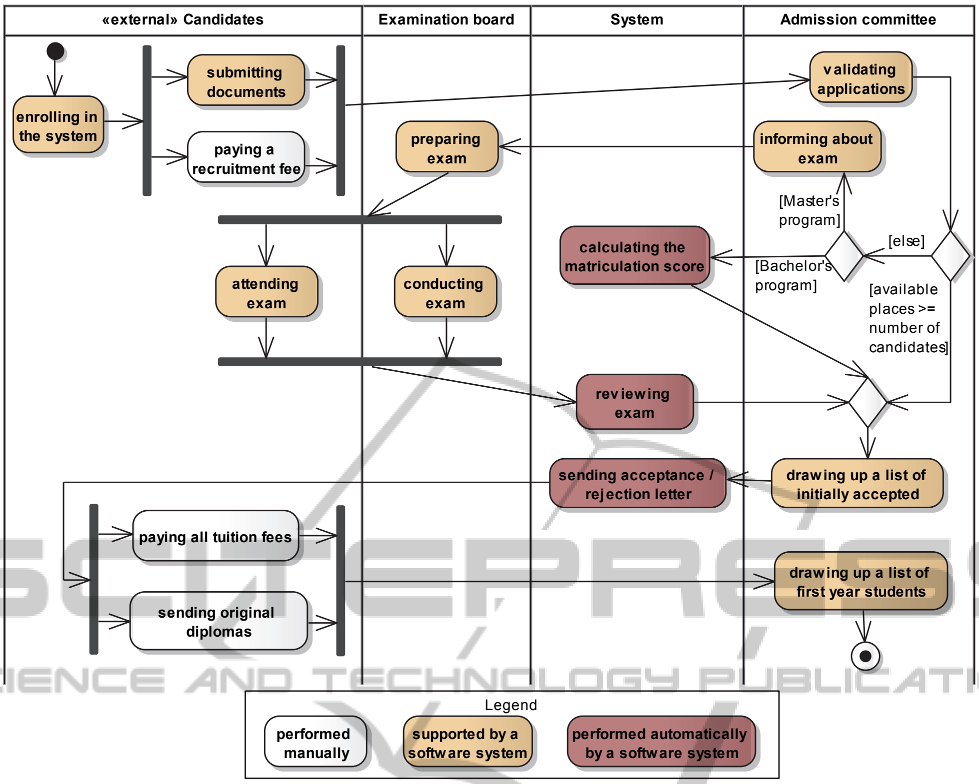

Figure 4: To-Be business model for the admission process.

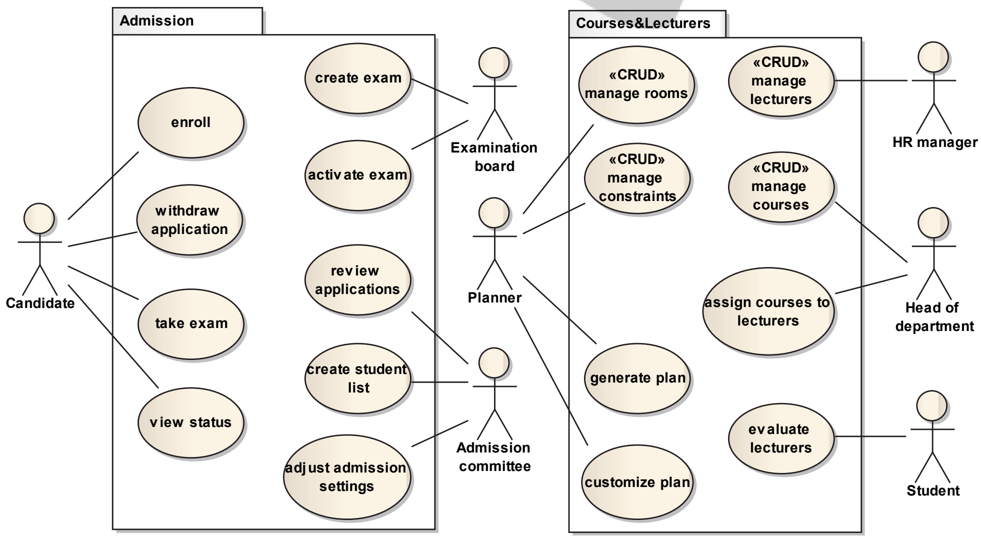

Figure 5: Use case diagram.

"M", we express this information using background color.

After accomplishing the To-Be business models, we applied the guidelines presented in Section 3.3 to derive use cases (Fig. 5). Next, we traced back the resulted use cases to their source. Table 3 lists all the EBPs consisting on the admission process down the left column. In the row across the top, it lists the use cases derived to support the stated processes. A traceability relationship is indicated with an X in the cell to record the fact that a specific use case has been defined for the purpose of supporting one or more business process. Note that a single process may be supported by multiple use cases.

Table 3: Traceability matrix for the admission sub-processes to use cases.

	enroll	withdraw application	take exam	view status	review applications	create exam	activate exam	adjust admission settings	create student list
enrolling in the system	x	x		x					
submitting documents	x								
validating applications					x				
informing about exam								x	
preparing exam						x			
conducting exam			x						
reviewing exam						x			
calculating the matriculation score					x			x	
drawing up a list of initially accepted									x
sending acceptance / rejection letter									x
drawing up a list of first year students									x
paying a recruitment fee									
paying all tuition fees									
sending original diplomas									

5 RELATED WORK

The most recognized technique that enhances the quality of the deliverables resulting from the requirements engineering process is the Use Case Driven Approach (UCDA) introduced by Jacobson (1987). Being use-case driven means that use cases bind together all software models. The basic concepts of UCDA are actors and use cases (Jacobson, 1992). An actor is a specific role played by a system user, and represents a category of users that demonstrate similar behaviour when using the system. The way an actor uses the system is described by use cases. A use case specifies a sequence of actions, including variants, which actor and system perform in order to yield observable results of value to a particular actor (Jacobson et al., 1999). UCDA helps to manage complexity, as it focuses on one specific aspect of usage at a time. It also provides means for customers and users to actively participate in requirements analysis, as use cases are expressed in terms familiar to them (Regnell et al., 1996). However, UCDA does not present any concrete heuristics to guide the requirements elicitation process and does not provide explicit means which could be used for linking business processes and use cases. These lacks may result in use cases without any underlying rationale. BORE improves the original UCDA by integrating it with business process engineering and by providing an alternative elicitation process, where use cases are derived from business process models.

As we mentioned before, BORE is designed on the basis of many existing techniques, ideas, guidelines and best practices in academia and industry as well as our experience from previous projects. Below, we briefly present the work that had the most impact on our approach.

Dijkman and Joosten (2002) compare the metamodels of use case diagrams and business process models. Then, they formally specify a procedure to transform business process models into use case diagrams, that complies to this mapping. Nevertheless, their procedure has some deficiencies. First, it assumes that all business processes are supposed to be computer-aided. However, some processes may simply be more suitable for being performed by hand. Therefore, in our approach the cost–benefit analysis precedes the derivation of use-cases and every elementary business process is labelled as “A”, “S”, or “M” according to the effect that the anticipated IS will have on it. Moreover, in our approach the derivation of use cases is just one step in the overall process that results in a use-case model. Second, their procedure does not cater for inclusion, extension or generalization of use cases. As a consequence, the obtained diagrams need manual reconstruction. Indeed, in the evaluation case study they conducted in the mortgage department of a bank, 40% of the generated use cases had defects. It seems that, even if improved, the transformation process cannot be done automatic because it always requires human intelligence.

A similar approach to (Dijkman and Joosten, 2002) can be found in the work of Odeh and Kamm (2003). They propose a method to explore relationship between business process model and use case model. They use Role Activity Diagram (RAD) to model business. Their method leads to the derivation of use case model from a process model. Thus, their and our approach have a similar goal, but we achieved this goal in a different way, using different techniques and notations.

The work of Odeh and Kamm (2003) is continued by Aburub (2012) who propose a four-steps method: develop a business process model using RAD model, identify automated activities, link each business objective with automated activities, and develop use case model based on objectives and automated activities.

Štolfa and Vondrák (2004) describe business process modeling as a tool for definition of requirements specification. They claim that there are repeatable situations during transition between business process modeling and requirement election. Next, they demonstrate three patterns that can be applied to support the transition between business process modeling and other phases of software process. These patterns can be considered as a complement to the third stage of our approach.

Vara et al., (2008) propose an approach that tries to prevent common mistakes detected in practice such as 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. Their approach is based on purpose analysis through BPMN and the MAP model. Map is a goal/strategy-driven approach to capture the intentions (goals) of an enterprise or system and determine the strategies that can contribute to the fulfilment of these intentions. Initial BPMN models (As-Is) are updated by the results of the analysis of the MAP model to get the To-Be business process models. The models are validated by end-users, and then analyzed in order to agree on the effect that the IS may have on the organizational needs. Finally, requirements are specified by means of the description of the business process tasks to be supported by the IS. Our main objection to their approach is that it introduces accidental difficulties by using three different notations (BPMN, Map and UML) during requirements analysis. Nevertheless, we borrowed the explicit distinction between As-Is and To-Be business models from their approach.

BORE also adopts several ideas from EKD (Bubenko et al., 2001). EKD is an approach that provides a systematic and controlled way of analyzing, understanding, developing and documenting an enterprise and its components. The EKD Enterprise Model comprises six interrelated sub-models that describe different aspects of the enterprise. Links between sub-models make the model traceable. In contrast to EKD which can be applied for many different reasons, our approach is subject to a single well-defined purpose. Therefore EKD provides means for modeling several aspects of an enterprise, while our approach concentrates all efforts on business process models and the way how these models can be used to support requirements elicitation.

6 CONCLUSIONS AND ONGOING WORK

Various methods and techniques have been proposed for helping make requirements precise, complete, and consistent. Each of them individually addresses some problems, but when arbitrary combined with others may introduce accidental difficulties. In this study, we demonstrated how to gain synergetic effect from combining methods and techniques from different fields of business engineering and requirements engineering.

The contribution of our research is twofold. On the one hand, it proposes a structured approach for deriving system requirements based on business process models. This approach is especially effective when system requirements are not fully knowable up front and must be discovered. Moreover, the built-in traceability supports impact analysis when an IS must be adapted to the evolving business processes. By providing seamless transition from business process modeling to use case modeling, our approach can be seen as a further step towards bridging the gap between business process engineering and requirements engineering. We also believe that the application of BORE can address the number of accidental problems and provide a stable foundation for attacking the essential difficulties. On the other hand, our study brings solutions to improve administrative efficiency of the real organization.

An action research evaluation builds on what already happens in projects and uses stakeholders' opinions. In our study, the stakeholders stated that BORE allowed them to better understand the business processes and, consequently, the requirements. Unfortunately, the opinions are flawed due to the following reasons: (1) the top management is satisfied because we are computerizing its organization for free; (2) potential users are satisfied because the anticipated software system is likely to lighten their work; (3) we are satisfied because we achieved our research objectives. Moreover, the opinions were obtained by discussing with the stakeholders informally. To overcome the mentioned weaknesses, we intend to employ BORE in commercial projects, design a form to survey these projects, and conduct an evaluation process using the framework defined in (Bobkowska, 2005). Future work is also oriented towards enriching the second stage of our approach with Business Process Simulation which is considered as one of the techniques suitable for discovering process bottlenecks and investigating business alternatives.

ACKNOWLEDGEMENTS

The study is funded by Polish National Science Centre, project DEC-2011/01/N/ST6/02752.

REFERENCES

Aburub, F., 2012. Activity-based approach to derive system models from business process models. In: *International Conference on Information Society*, London, UK

Bendakir, S., El Hamlaoui, M., Zarour, N., 2013. An agent-oriented approach to support requirements prioritization processes. In: *IEEE International Workshop on Requirements Engineering 2013*, Constantine.

Bobkowska, A., 2005. A framework for methodologies of visual modeling language evaluation. In: *Proc. of the 2005 symposia on Metainformatics*, Esbjerg, Denmark.

Brooks, F., 1987. No Silver Bullet: Essence and Accidents of Software Engineering. In: *Computer 20*, 4 (April 1987), pp. 10–19.

Bubenko, J. A., Persson, A., Stirna, J., 2001. EKD User Guide, Royal Institute of Technology (KTH) and Stockholm University, Stockholm, Sweden.

Dijkman, R. M., Joosten, S. M. M., 2002. An Algorithm to Derive Use Cases from Business Processes. In: *6th IASTED International Conference on Software Engineering and Applications*, Cambridge, MA.

Eriksson, H., Penker, M., 2000. Business Modeling with UML: Business Patterns at Work. John Wiley & Sons.

Faulk, S., 1997. Software Requirements: A Tutorial. In: Thayer, R., Dorfman, M. (Eds.): *Software Requirements Engineering. IEEE Computer Society* press.

Frost, S., Allen, P., 1996. Business Process Modeling, *SELECT Software Tools plc*, Cheltenham

Gawin, B., Marcinkowski, B., 2013. Simulating Business Processes. *BPMS and BPMN Standards Applied* (in Polish). Helion.

Gotel, O., Finkelstein, A., 1994. An Analysis of the Requirements Traceability Problem. In: *1st International Conference on Requirements Engineering*, Colorado Springs, CO.

Górski, J., 2000. Inżynieria oprogramowania: w projekcie informatycznym (in Polish). MIKOM.

Hammer, M., 1990. Reengineering work: don't automate, obliterate. *Harvard Business Review*, July-August 1990, pp. 104–112.

Harmon, P., Wolf, C., 2011. Business Process Modeling Survey. *Business Process Trends.*

Jacobson, I., 1987. Object-Oriented Development in an Industrial Environment. In: *SIGPLAN Conference on Object-Oriented Programming Systems, Languages, and Applications (OOPSLA'87)*, Orlando, FL.

Jacobson, I., 1992. Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley.

Jacobson, I., Booch, G., Rumbaugh, J., 1999. The Unified Software Development Process. Addison-Wesley.

Leffingwell, D., Widrig, D., 1999. Managing Software Requirements. Addison Wesley.

Lehtola, L., Kauppinen, M., Vähäniitty, J., Komssi, M., 2009. Linking business and requirements engineering: is solution planning a missing activity in software product companies?. In: *Requirements Engineering* vol. 14(2), April 2009, pp. 113–128.

Maciaszek, L., 2005. Requirements analysis and system design. Addison-Wesley.

Monsalve, C., April, A., Abran, A., 2010. Representing Unique Stakeholder Perspectives in BPM Notations. In: *8th ACIS International Conference on Software Engineering Research, Management and Applications*, Montreal, Canada.

Marcinkowski, B., 2010. Applying Business Process Modeling Techniques: Case Study. *Journal of Internet Banking and Commerce 15*, Array Development.

Odeh, M., Kamm, R., 2003. Bridging the gap between business models and system models. In: *Information and Software Technology*, (45)15, pp. 1053-1060.

Przybyłek, A., 2007. The Integration of Functional Decomposition with UML Notation in Business Process Modelling. In: *Advances in Information Systems Development*, Vol 1, pp. 85–99.

Przybyłek, M., 2013. Skeletal Algorithms in Process Mining. In: *Studies in Computational Intelligence*, vol. 465, pp. 119–134.

Przybyłek, M., 2013. Process mining through tree automata. In: *5th International Conference on*

Evolutionary Computation Theory and Applications, Vilamoura, Portugal.
Przybyłek, M., 2013. Dynamic data discovery. In: *8th International Conference on Knowledge, Information and Creativity Support Systems*, Cracow, Poland.
Ravichandar, R., Arthur, J. D., Perez-Quinones, M., 2007. Pre-requirement specification traceability: Bridging the complexity gap through capabilities.
Regnell, B., Andersson, M., Bergstrand, J., 1996. A Hierarchical Use Case Model with Graphical Representation. In: *IEEE International Symposium and Workshop on Engineering of Computer-Based Systems*, Friedrichshafen, Germany.
Robertson, S., Robertson, J., 2013. *Mastering the Requirements Process*. Addison Wesley.
Sjoberg, D., Dyba, T., Jorgensen, M., 2007. The future of empirical methods in software engineering research. In: *2007 Future of Software Engineering (FOSE'07)*, Minneapolis, MN.
Štolfa, S., Vondrák, I., 2004. A Description of Business Process Modeling as a Tool for Definition of Requirements Specification. In: *12th International Conf. on Systems Integration*, Prague, Czech Republic.
Taylor-Cummings, A., 1998. Bridging the user-IS gap: a study of major information systems projects. *Journal of Information Technology 13*, pp. 29–54.
Torkar, R., Gorschek, T., Feldt, R., Svahnberg, M., Raja, U., Kamran, K., 2012. Requirements Traceability: a Systematic Review and Industry Case Study. *International Journal of Software Engineering and Knowledge Engineering 22*(3), pp. 385–434
Vara González, J. L., 2011. Business process-based requirements specification and object-oriented conceptual modelling of information systems. PhD thesis, Universidad Politecnica de Valencia.
Vara, J.L., Sánchez, J., Pastor, Ó., 2008. Business Process Modelling and Purpose Analysis for Requirements Analysis of Information Systems. In: *20th international conference on Advanced Information Systems Engineering*, Montpellier, France.
Weerakkody, V., Currie, W., 2003. Integrating business process reengineering with information systems development: issues & implications. In: *2003 international conference on Business process management (BPM'03)*, Eindhoven, Netherlands.
Wolf, C., Harmon, P., 2012. The State of Business Process Management. *Business Process Trends.*
Wrycza, S., Marcinkowski, B., Wyrzykowski, K., 2005. UML 2.0 in *Information Systems Modeling* (in Polish). Helion.