Using Data-Centric Business Process Modeling for Discovering Requirements for Business Process Support Systems: Experience Report

Ilia Bider[1,2], Erik Perjons[1], and Zakria Riaz Dar[1]

[1] DSV, Stockholm University, Stockholm, Forum 100, SE-16440 Kista, Sweden
[2] IbisSoft AB, Stockholm, Box 19567, SE-10432 Stockholm, Sweden
ilia@{dsv.su,ibissoft}.se, perjons@dsv.se, zrdar@kth.se

Abstract. Building a process model is a natural part of the requirements engineering (RE) when creating requirements for a computerized system/service to support a business process. When the process in question is workflowable (i.e. a process in which the order and the flow of tasks/operations/activities can be predefined), there are plenty of modeling techniques, notations and tools that can help in this undertaking. These techniques, however, are of little use for discovering requirements for support of non-workflowable processes in which the information artifacts created in the process (e.g. reports, lecture slides, budget documents) are of more importance than the flow of tasks/operations/activities. Other types of techniques, notations and tools are required in this case. This paper reports on a project of using a data-centric modeling approach supported by a computerized tool in RE. The goal of the project was to test whether the approach could be useful for the task of discovering requirements on a computerized system/service supporting the process, and which and how much of requirements could be captured using it. The process used in the test is a process of course preparation in the authors' own department. The paper reports on the environment in which the project has been conducted, results achieved, and lessons learned.

Keywords: Requirements Engineering, RE, Requirements discovery, business process, data-centric.

1 Introduction

Following Ian Alexander [1], we consider that all important requirements cannot be gathered from stakeholders directly, but need to be discovered, which warrants using special techniques and tools different from the ones used for managing already discovered requirements. As our concern is requirements for computerized systems/services that support business processes, discovering details of the process to support is an essential part of the requirements discovery.

A systems/services can be aimed at supporting an already existing process, or a process that needs to be designed or improved alongside with developing a support system. Independently of which of the above is the case, it is people who are (will be)

S. Nurcan et al. (Eds.): BPMDS 2013 and EMMSAD 2013, LNBIP 147, pp. 63–77, 2013.

engaged in the process who have relevant tacit knowledge that needs to be unearthed during the discovery of requirements. Therefore, the role of techniques and tools used in the requirements discovery is to facilitate the existing or future process participants to reveal the tacit knowledge they possess. According to [1], techniques and tools for requirements discovery should be quite simple so that the focus will not be moved from discovering requirements to designing the system.

When there is a good chance that the process to be discovered has a strict order of tasks/operations/activities, usual process modeling techniques based on the workflow-thinking could be tried in the discovery process. These range from simple charts to complex workflow diagrammatic languages such as BPMN 2.0, and they are supported by a number of modeling tools. However, when the chances that the process will be workflowable[1] are small, these techniques and tools might not be appropriate, and other means should be engaged in the requirements discovery phase.

In this paper, we consider the problem of discovering requirements for processes in which information/data processing by collaborative teams constitutes the core of the process. In addition, we do not require such process to be workflowable. We believe that for this kind of processes, a data-centric process modeling technique is more appropriate as far as process discovery is concerned.

In this paper, the term data-centric modeling is understood in a broad meaning. Namely, as data-centric we consider any process modeling technique that permits to start structuring data/information processed in the frame of the process before the details of the flow of tasks/operations/activities are known. To this category, for example, belong artifact-based modeling [3], data-driven modeling [4] and state-oriented modeling [5]. Defining folder structures for case-based systems [6] could also be considered as belonging to the data-centric process modeling[2].

The goal of the project reported in this paper was to investigate whether a data-centric modeling technique supported by a computerized tool is suitable as a means for discovering requirements for business process support (BPS) systems/services. More specifically, we aimed at getting answers to the following **three questions**:

1. Whether such an approach is suitable for use in requirements discovery facilitating workshops.
2. Whether the requirements discovered in the workshops could be represented in a form suitable for discussing them with a broader audience that includes stakeholders who have not participated in the facilitating workshops.
3. Which and how much of requirements could be discovered with this approach .

Our search of the works related to the above questions produced no results, thus, to the best of our knowledge, the current work is the first attempt to get answers to these questions[3].

---

[1] As workflowable, we consider a process where the order and the flow of tasks/operations/activities can be predefined. For more exact definition of workfloability see [2].

[2] The main difference between a data-centric and traditional workflow process modeling is that in the former the focus is on information artifacts, e.g. reports, lecture slides, budget documents, while in the latter the focus is on operations/activities that produces the artifacts.

[3] Our past experience of state-oriented process modeling [4] lacked proper tool support.

Investigation was conducted in the frame of a real organization – department of Computer and Systems Sciences at Stockholm University. Though the study has been conducted only in one organization, there is a likelihood that the results achieved are of general nature; based on the authors' previous experience (see some examples in [5]) this particular environment is quite typical for non-workflowable processes.

The rest of the paper is structured according to the following plan. In Section 2, we describe the project settings that include short description of the organization, business process under investigation (course preparation process), and project team. Section 3 describes our efforts to find a tool to use in the project. Section 4 overviews the tool used in the project. Section 5, describes completion of the project. Section 6 discusses the model built during the project. Section 7 discusses the results achieved and lessons learned. Section 8 contains concluding remarks and plans for the future.

2 The Project Environment

2.1 The Organization

The project has been completed in the department of Computer and System Sciences, abbreviated to DSV, at Stockholm University. The department is engaged in research and undergraduate and graduate teaching of about 5 700 students simultaneously. It runs bachelor, master, and doctoral programs in the fields of Computer Science and Information Systems. It has about 280 staff members including teachers and administrative personal. It also has its own IT department that operates department specific software, while the general software is operated by the central IT unit of Stockholm University.

The IT department, besides operating the software acquired from various vendors, has its own development unit engaged in developing department specific software. The latter includes development of systems that supports teaching and learning. The unit does not have a strong tradition of requirements engineering, which results in long cycles of getting the system and it users synchronized.

Modern process modeling tools are not used during requirements engineering phase in the department. The systems that support teaching and learning, from outside vendors and from own development, are not of the type of process aware systems. They have quite reach functionality, but the information on when and how to use the functions included in these systems resides mostly in the heads of their users.

2.2 The Process

The business process chosen for the study is the process of preparing a course occasion to be given by the department. The occasion can be the first occasion of a completely new course, or just an ordinary occasion of a regularly given course. This business process has been chosen based on the following two reasons:

- It is a typical process in the department.
- The process does not have real computerized support. The results of it are to be placed in different systems, e.g. lection slides needs to be made available to the students for download. However, these systems do not support course preparation.

Below, we present an informal overview of the chosen process. In this overview, we also include activities related to giving and evaluating the course. We identify five major phases in the course process[4]:

1. *Planning course* includes a number of meetings with involved teachers to decide which teaching and learning activities to carry out during the course as well as their sequence. The phase also includes deciding and producing the course material for the course. Finally, an evaluation form needs to be designed, which will be filled out by the students when the course has ended.
2. *Schedule course* consists of composing a schedule with dates, times and locations for the lectures, lessons and seminar as well as a date, time and location for the written exam and other teaching and learning activities. The phase includes a number of interactions between the teacher responsible for the course and the person responsible for scheduling courses in the department.
3. *Publishing course material* consists in printed course material. The printing is done by the person responsible for printing.
4. *Learning and teaching* includes a number of teaching and learning activities, such as lectures, lessons and seminars, managing assignments and carrying out exams. It also includes giving feedback on and/or grading reports and exams.
5. *Evaluation* includes the students evaluating the course after the end of the course. The phase also includes an analysis of the evaluation carried out by the teacher responsible for the course.

2.3 The Team

The project involved four teachers and one MS student; this group will be referred to as the *extended group*. The major team consisted of two teachers and one MS student (all authors of this paper); this group will be referred to as the *modeling group*. One of the teachers had long experience of giving courses in the department, the other one was a novice. The student represented the "learning" stakeholders. The additional two teachers, referred to as the *external domain experts,* had long experience of teaching in the department. They participated only in the evaluation of the approach's results. They were not involved in requirements discovery, and knew nothing about the project beforehand.

3 Selecting a Modeling Tool

3.1 Requirements on a Tool to Be Used

We were looking for a data-centric process modeling tool or a BPM suite of this kind that could be suitable for performing multiple Requirements Engineering (RE) tasks

[4] Though the process is split in a number of phases, the latter are not being executed in a sequence but can run in parallel (see Section 6), which makes the process non-workflowable according to [2]. Full analysis of workflowability of this process is not presented due to the size limitations, but will be published elsewhere in connection to another topic.

[1]. In particular, we looked for a tool that could be used directly in facilitating workshops for discovering the following aspects of the process to be supported:

1. *Structure of data/information* created and utilized in the process
2. *Data/information flow* in the process
3. *Participant collaboration* in the frame of process instances
4. *Categories of users* engaged in the process and limitation on the data/information they can access (read, write, or modify)
5. *Operations/activities* included in the process and *restrictions* on the order in which they can be completed

Additionally, we preferred a tool/suite to be suitable for:

– Designing a *prototype* of the system to give future users some understanding of what would it mean for them to run a process supported by the system to-be
– Discussing and recoding process *scenarios* based on the past experience (process cases)

In addition, we preferred a domain independent tool that could be used for various kinds of processes, general administration, teaching and learning, research, etc.

3.2 Searching for a Tool

Right from the project start we had one candidate for a tool to be used in the project, namely, a cloud-based service called *iPB* [7,8], developed by *ibisSoft* with which the first author was associated. *IbisSoft* had the policy of providing a limited demo-license for research purposes free of charge, so it was easy to obtain access to the tool. Though *iPB* had not been explicitly developed as a data-centric modeling tool/suite, our preliminary analysis showed that *iPB* satisfied the requirements set in the previous section.

Despite having a candidate, we decided to spend some time looking for other candidates to be used, in case we can find a better alternative. In preparation for the tool selection, we created a list of criteria for tool evaluation. This list is based on the requirements from the previous section and general properties of modeling tools from the literature, see, for example, [9]. The list includes the following criteria:

– *Availability*. Firstly, the tool should be available for usage, e.g. commercially available, or as an open source. Secondly, it should be easily accessible from any place one can possibly need to have access to it [1]. For our purpose, it would be desirable to have a web-based/cloud-based tool.
– *Domain-independence* (expressiveness or universality in terms of [9]). The tool should be possible to use in different application domains.
– *Completeness* [9]. The tool should have means to express all concepts considered to be important for the given objective of modeling.
– *Comprehensibility* [9]. The models, even the intermediate ones, should be easy to comprehend for domain specialists without prior knowledge of business process modeling.

– *Tasks flexibility*. The tool should be possible to use for different types of activities like modeling, prototyping, scenario testing (see the list in the previous section).
– *Tasks suitability* - extension of task flexibility. The tool should be suitable for the tasks for which it has been chosen [9], i.e. allows completing them in a convenient way.
– *Usage flexibility* (or arbitrariness in terms of [9]) – extension of task suitability. The tool should not impose hard restriction on its usage, but gives the user freedom to choose how to use it.
– *Coherence* [9]. Different components produced with the help of the tool should be integrated to constitute a whole.

When searching for a tool, we did not have in mind finding the best possible candidate that satisfies the criteria, the first good enough choice would be sufficient for us. The goal of the whole project was to test a data-centric approach of business process modeling supported by a tool/suite. Which tool to use was considered of lesser importance.

Through the quick search on "data-centric" and "artifact-based" process modeling we found a number of research articles, but only two references to potential tool candidates, both from IBM - FastPast and Siena described in [3]. Both were experimental tools that were supposed to be available for research and education purposes, but we found no URL with instructions on how to get access to them.

As our initial efforts to find an available tool through a general search were unsuccessful, we decided to stop the search and use the tool already at our disposal.

4 The Modeling Tool Described

4.1 *iPB* as a Data-Centric Business Process Modeling Tool

iPB [7,8] was designed as a tool for developing BPS systems/services for loosely structured business processes. One part of such development consists of designing a process model that the tool interprets at runtime while providing support for process participants.

iPB consists of two components - *Design studio*, and *Runtime environment*. Design studio is used for building a process model, while Runtime interprets this model helping process participants to run their process instances/cases.

Process modeling in *iPB* is based on four *main* abstract concepts: *Process map, Process step*, *Step form*, *Form field* (additional concepts are described later)The basic relationships between these *main* concepts are as follows.

– A *process map* consists of a collection of named *process steps* arranged on a two-dimensional surface called *process layout*. The layout consists of two areas – (a) the upper row called *flow-independent* steps, and (b) a low area, a two dimensional matrix called *flow-dependent* steps, see Fig. 1.
– Each *process step* in a *process map* has a *step form* attached to it.

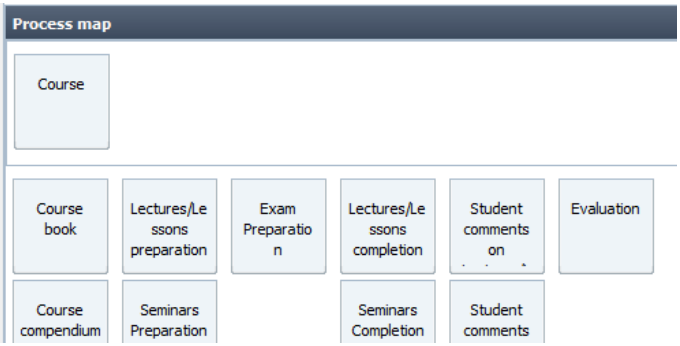

Fig. 1. Process map

– A *step form* consists of a collection of named *form fields* arranged in a two-dimensional matrix called *form layout*, see Fig. 2. Each field belongs to a certain type. There are simple types, like, text, multi-text, date, date-time, option list, checkbox (boolean), etc., and complex types, like uploaded file, journal, person, organization. In addition, field collections that belong to different step forms can intersect. This is done by defining fields in one form, and then referring to them in other forms.

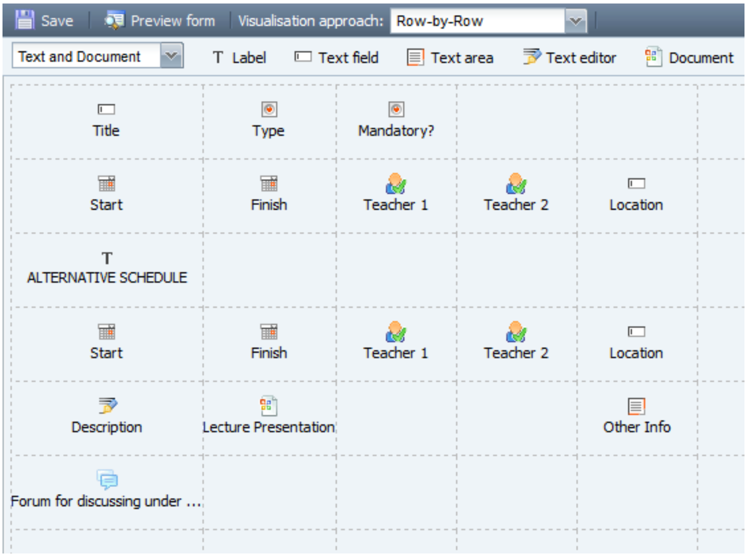

Fig. 2. Step form for step Lectures/Lesson preparation from Fig. 1

A process map with underlying step forms defines a kind of a database scheme for the given process type/template. This scheme is then used by the runtime for "creating" a database for each instance/case of the process, started based on this process map. The runtime system also interpret step forms as web forms for inputting, viewing and changing information in the instance database, see Fig. 3. The process map is used for creating an *instance map*, see Fig 4, that serves as a mechanism for user navigation through the web forms. To open the web-form, the user just clicks on the step in the *instance map*. As we see from Fig. 3, some steps are allowed to have multiple forms at runtime (see the tabs for each lecture in Fig, 3).

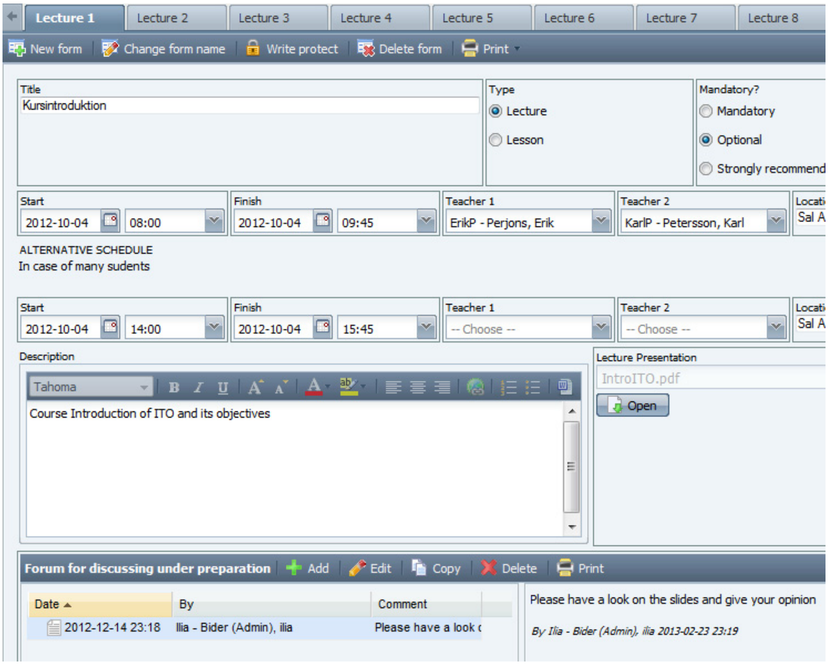

Fig. 3. Step form from Fig. 2 as a web-form

Besides the *main* concepts described above, *iPB* provides additional capabilities, e.g.

- *Business rules* that establish relationships between the steps. One type of such rules controls whether the user can open a particular step form based on the state of completion of other forms. Steps that cannot yet be clicked on are colored grey, see Fig. 4. Such rules are specified with the help of a square matrix where both rows and columns correspond to the steps defined in the process. In this matrix, the content of a cell can determine that the row step can be started only after the column step has been completed (blue color in Fig. 4), or started (green color in Fig. 4). Other types of rules prescribe synchronization of steps with multiple forms, e.g., steps "Lectures/Lessons preparation" and "Lectures/Lessons completion" in Fig. 1 and 4 (for more on synchronization see Section 6).
- *Business rules* that establish when data in a step form can be saved or the step can be closed. Such rules are expressed via defining some fields on the form as mandatory for save, or close.
- *User profiles* that specify categories of users and their access rights to read, and change information in each step form.
- *Visual properties* assigned to the fields that change their look and feel at runtime.
- *Step properties*, such as permission to have more than one form (see Fig 3).

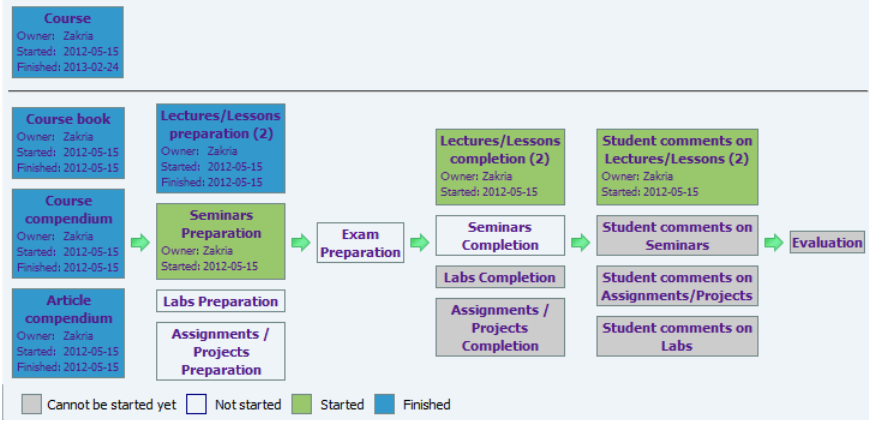

Fig. 4. Instance process map from Fig. 1 as a user navigation mechanism

4.2 Business Process Modeling with *iPB*

In section 3.1, we listed aspects of a business process that we wanted to be depicted in the model, and additional tasks that the modeling tool chosen for the project could help to perform. Below, we show how these aspects and tasks are covered by *iPB*.

- *Structure of data/information* created and utilized in the process is defined through creating step forms.
- *Data/Information flow* in the process is defined by including references to the form fields from the previous step forms into the form where this information is used.
- *Participant collaboration* in the frame of process instances is defined by inserting complex fields of the type *Journal* in the step form; see the field at the bottom in Fig. 2 and 3. Users working with the same form add records to the journal registering questions, answers, comments, etc. relevant to the particular step.
- *Categories of users* engaged in the process and limitation on the data/information they can access (read, write, or modify) are defined with the help of user profiles.
- *Tasks/operations/activities* included in the process and *restrictions* on the order in which they can be completed are specified in the following manner. The concept of step is used for representing larger chunks of work - work-packages; if smaller operations are needed they are represented in the step forms as checklists in the way as accepted for case-management/adaptive case management systems [6]. The order of steps is represented by (a) process layout that gives a recommended sequence of steps (from left to right and from top to bottom), and (b) by business rules that restrict the possibility to "jump over" some steps. On the lower level, making operation checklists mandatory gives an opportunity to prevent finishing the step to which they belong until all operations have been completed. The latter can prevent starting the next step if it is forbidden by business rules that require finishing the previous step first.

- As far as *prototyping* is concerned, the *iPB*'s runtime system automatically creates a system prototype that can be tested by the future end-users. By using visual properties of fields, there is a possibility even to design the exact layout of the forms to be used in the future system.
- The run-time system allows also to record and discuss *scenarios* of the past process instances/cases.

Note that the main difference between *iPB* modeling principles and that of other data-centric modeling tools [3,4] is that *iPB* does not use workflow notation for describing the flow of work, which is the case with other tools. The order, when needed, is determined by various kinds of business rules.

5 Building and Demonstrating a Model

The project presented in this paper included the following activities (a) three facilitating workshops were all members of the *modeling group* met for brainstorming discussions, (b) modeling work between the workshops based on the discussions and available materials, (c) presentation to and discussion with the *external domain experts*[5] who did not belong to the modeling group, and (d) writing a report.

The main bulk of the domain knowledge needed for the project came from the modeling group's own experience. Additional knowledge came from a study of the existing systems used for conducting teaching in the department at the time. The traces of the past occurrences of courses in these systems were used to record and discuss scenarios of how these occurrences could look like in the new system.

The first facilitating workshop consisted of informal discussions, the result of which was a general description of the course preparation process presented in Section 2.2 which was made in a *phase-flow* manner. In the second facilitating workshop, the first draft of a *data-centric* process model had been developed, see Fig. 5. This draft was then extended by designing detailed step forms to some steps in Fig. 5; this was done before the third facilitating workshop. The third workshop discussed the first draft by running a scenario of a recently completed course occurrence. Based on this discussion, a list of changes was agreed upon; some of the changes were directly made in the *iPB* model during the workshop. After the workshop, the model was changed according to the list and got the form of Fig. 6. Additionally a relatively full scenario of a past case was recorded using the *iPB* runtime system. Details of the final model are overviewed in Section 6.

After completing the changes, the model was presented to the *external domain experts* (both teachers). The presentation consisted of showing the model in the run-time environment, see Fig. 4 and 3. This was done first by showing how to start preparation of a new course occurrence, and then going through the steps of already recorded scenario. Then the *extended group* (*modeling group* + *external domain experts*) discussed which requirements on the support system where properly discovered and which were left outside the model and the prototype demonstrated. The conclusions reached in this discussion are overviewed in Section 7.

The results of the project were reported as master thesis written according to the design science research principles.

---

[5] For working definitions of terms *modeling group* and *external domain experts* see Section 2.3.

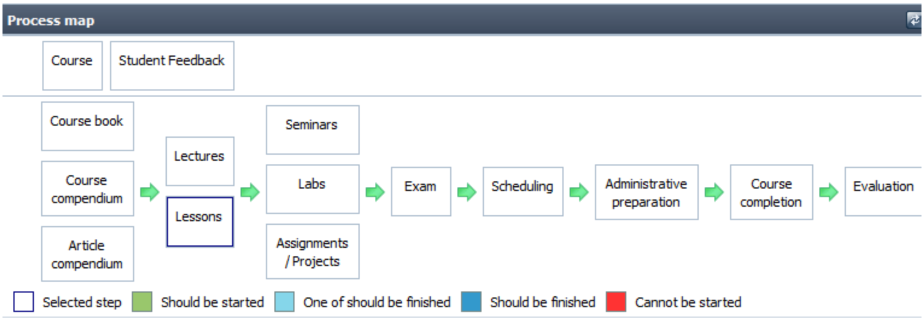

Fig. 5. The first version of the course preparation model

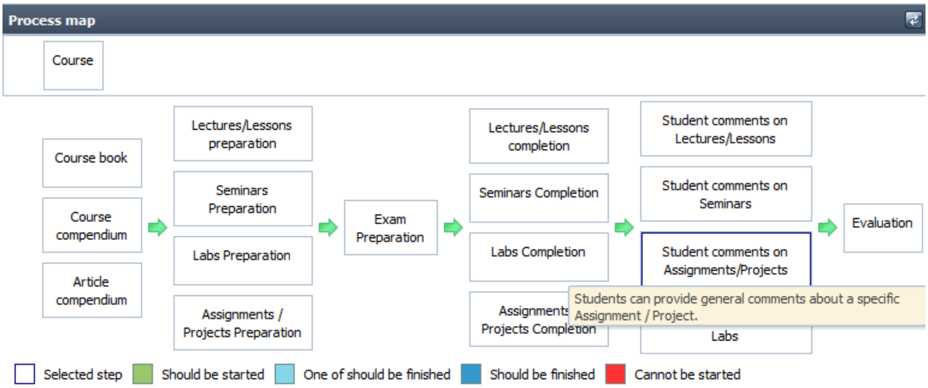

Fig. 6. The final version of the course preparation model

6 The Model Described

The final process map is presented in Fig. 6, Fig. 1 (process map layout in the design studio) and Fig. 4 (an instance process map at runtime). Additionally, Fig. 2 and 3 demonstrate a step form typical for this process, more exactly the form for step "Lectures/Lessons preparation". Fig. 2 gives a step form view in the design studio and Fig. 3 at runtime. As follows from Fig. 6, the process map is clearly divided in two parts: (1) preparation of the course on the left (up to and including *Exam preparation*), and (2) following up the course.

The first part is structured according to the artifacts that need to be prepared for the course, each component being represented as a step in the model (see Fig. 6). This part includes (a) general components: *Course* (description), *Course book* (optional), *Course compendium* (mandatory), *Article compendium* (optional), *Exam preparation*, and (b) specific components that correspond to teaching and learning activities: *Lectures/Lesson preparation, Seminar preparation, Labs preparation,* and *Assignments/Projects Preparation*.

Each step has a step form attached that defines the data structure for this step. A typical form, the one for step *Lectures/Lesson preparation*, is presented in Fig. 2 (the design studio view), and Fig. 3 (runtime view). It includes the descriptive fields, like: *Title*, *Type*, *Description*, *Lecture presentation*, and the administrative fields, like *Teacher*, *Start, Finish, and Location*. In addition, it has a journal like widget *Forum for discussing under preparation* were the teachers preparing this teaching/learning activity can leave comments for themselves and for each other. Note that the form in Fig. 3 allows multiple forms at runtime (see tabs) – one for each activity. Multiple forms are also allowed for all other steps except *Course*, *Course compendium*, *Exam preparation* and *Evaluation*.

The system for which the project was discovering requirements was aimed only for course preparation and evaluation; activities for giving the course were left outside as they were more or less covered by the existing systems. The second part of the model, all the steps on the right of step *Exam preparation*, deals exclusively with getting feedback for course evaluation from the teachers and students. More exactly, the steps the names of which end with "completion" are aimed at gathering feedback from the teachers immediately after the corresponding teaching/learning activity. The steps the names of which start with "Student comments on" are aimed at gathering feedback from the student immediately after the corresponding teaching/learning activity. This feedback is aimed to help in step *Evaluation*. An example of a feedback form is presented in Fig. 7. In it, journal field *Students Comments* is designated for recording feedback from students attending the course.

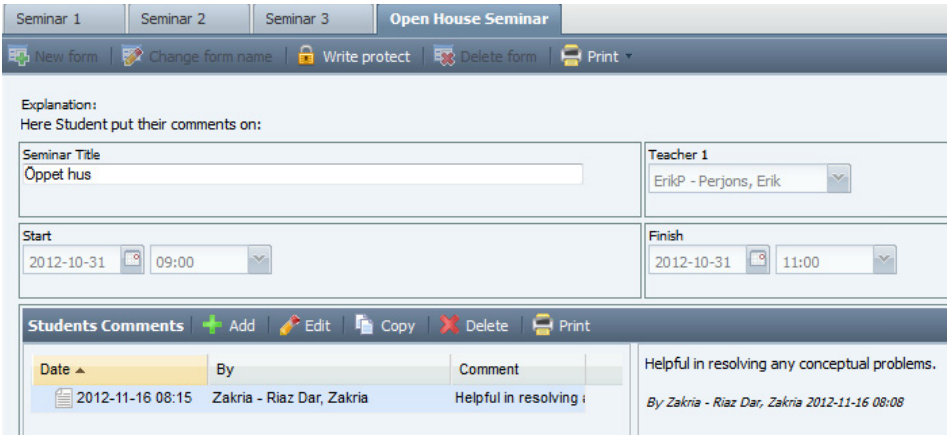

Fig. 7. Step form for Student comments on Seminars

The feedback steps are synchronized with corresponding preparation steps. When step A is synchronized with step B, the former will automatically get the same number of forms as the latter. If there are references on the A's form to the fields on the B's form, then the references are also synchronized so that they show the contents from the corresponding forms. For example, fields *Seminar title*, *Teacher 1*, *Start* and *Finish* in Fig. 7 are references to the fields on step form *Seminar preparation*, and show their content from the form *Open House Seminar*.

The order of steps in the model is mostly given as a recommended order according to the layout. We found that it is almost impossible to establish a strict order to which all teachers would abide. Though it is highly recommended to have all materials ready before the start of the course, it is happened that changes in some materials and even in the schedule are done very late, when the course has already been started. The business rules where used mainly in its "softest" form – some steps cannot be started unless some other ones has been already started, see their effect at runtime in Fig. 4.

As far as categories of users are concerned, we differentiated two categories: teachers and students. Teaches can access all steps, except that they cannot change any data in the steps designated for gathering feedback from the students. The students will need access to the latter, but not the former except the step *Course* that includes the general information on the course.

7 Analysis of Results and Lessons Learned - Summary

Material presented in this section is based on[6]:

– Own reflections of the *modeling group* (the three authors) over their experience from the project. This is used to answer the *first question* from Section 1, namely, suitability of data-centered process modeling supported by a computerized tool for direct usage in facilitating workshops aimed at discovery requirements.
– Interviews with the two *external domain experts* who were presented the results of our work, but who did not participate in the facilitating workshops. This is used to answer the *second question*, namely, suitability of the approach for presenting the results to the broader audience.
– Protocol of the brainstorming discussion of the *extended group* (*modeling group* + *external domain experts*). This is used to answer the *third question*, namely, how much of requirements could be discovered using a data-centric modeling approach.

Question 1. We came to the positive answer when considering the following *self-reflections*:
– It was relatively easy for us to start modeling in a data-centric fashion. The most important thing to do was to switch the focus from the task flow as described in section 2.2 to the results to be achieved. In our case, the latter was information artifacts to be prepared in the process, compendium, lectures slides, etc.
– Using data-centric approach supported by an appropriate tool inspired our creativity during the facilitating workshops. This was due to highly visual way of representing data-structures as web forms, and possibility of recording past cases. For example, during the third workshop, we discovered that initial presumption that each lecture requires only one teacher and one room does not correspond to the practice accepted in the department. A lecture can be given more than once in the frame of the same course occasion and by different teachers. During the same workshop, we came to the idea of introducing students and teachers feedback gathering during the course, instead of doing it after finishing the course.
– We found it quite convenient to hold discussions on data structures separately from those that concern establishing restrictions on the tasks/operations/activities flow.

[6] Due to the size limitations, only the summary of results is presented in this paper.

Question 2. We came to the positive answer based on the positive responses from the *external domain experts*. Namely, using *iPB* runtime environment constitutes an efficient way of presenting requirements discovered in the facilitating workshops due to the highly visual means for:

– representing data structures as web forms, see Fig 3.
– representing restrictions on sequence of tasks as grey colored boxes that are un-greyed when situation changes, see Fig. 4.
– recording past cases that were easy to follow by domain experts who were not included in the requirements discovery project.

In addition, the experts appreciated the functioning system prototype provided by the *iPB* runtime environment when it interpreted the process model. It gave quite good understanding of how a process aware system that supported course preparation could look like.

Question 3. During the brainstorming in the *extended group* (*modeling group* + *domain experts*) that directly followed the presentation, the consensus was reached that:

– All five types of requirements that we aimed to capture in Section 3.1 were indeed captured in the model to the degree sufficient for starting the system development. To these belong (1) data/information structures, (2) data/information flow, (3) participant collaboration, (4) categories of users, (5) operations included in the process and restrictions on the order in which they can be completed.
– Some requirements that could be of importance where not captured at all. In the first place, this comment concerns requirements on the needs and possible ways of integration with already existing systems. In the second place, this comment concerns the representation of stakeholders' goals. While the first comment is of importance and need to be dealt with in the future, the second one was outside the scope of the project.
– In addition, the open question remains whether an approach taken in the current project can be as good for more complex processes. This comment warrants additional testing.

As far as using *iPB* as a data-centric modeling tool is concerned, our experience shows that it satisfies the criteria listed in section 3.2 sufficiently to be useful in this kind of projects (more detailed analysis of this is not presented due to the size limitations). However, a more visual means for presenting information flow than just field references would be of help in such cases of the tool usage.

8 Concluding Remarks

As was stated in the introduction, the goal of the project was evaluating the suitability of a data-centric business process modeling supported by a tool for requirements discovering. The goal was fulfilled by actually building such a model for a representative process with the help of a tool, and presenting it to the stakeholders. The analysis from Section 7 shows that the approach is valid, but needs further testing and improvement, which will be included in our plans for the future. In particular, new testing would concerns using (a) other data-centric process modeling tools, (b) other business processes, and (c) other modeling and domain experts teams.

Our analysis also shows that having a tool that supports scenario recording and prototyping is important. We feel that without these features, the usefulness of the approach we suggest will be limited.

We also believe that our experience report could be of interest for a wider audience than the one that is interested in requirements discovery only. Data-centric process modeling is a relatively new area, and there is not that much experience on its usage reported in the literature. In addition, there is a lack of easily available tools for data-centric business process modeling (see Section 3.2). Therefore, the example and discussions presented in this paper may be of use for any researcher or practitioner interested in non-workflow process modeling. Furthermore, to the best of our knowledge, there is no accepted definition of what data-centric process modeling means and how it differs from other types of process modeling, in particular, artifact-oriented, data-driven, and state-oriented modeling. Discussion on this issue presented in Section 1 may serve as a starting point for clarifying the term and its relationship to other types of modeling.

Acknowledgements. Many thanks to IbisSoft and *iPB* development team – T. Andersson, A. Striy and R Svensson - for providing us with the tool. The authors are also grateful to our colleagues who served as domain experts: M. Henkel and J. Snygg, and to the anonymous reviewers whose comments helped to improve the text.

References

[1] Alexander, Dukic, L.B.: Discovering Requirements: How to Specify Products and Services. Wiley, UK (2009)

[2] Bider, I., Johannesson, P., Perjons, E.: Do Workflow-Based Systems Satisfy the Demands of the Agile Enterprise of the Future? In: La Rosa, M., Soffer, P. (eds.) BPM 2012 Workshops. LNBIP, vol. 132, pp. 59–64. Springer, Heidelberg (2013)

[3] Cohn, D., Hull, R.: Business artifacts: A data-centric approach to modeling business operations and processes. IEEE Data Eng. Bull. 32(3), 3–9 (2009)

[4] Müller, D., Reichert, M., Herbst, J.: Data-driven modeling and coordination of large process structures. In: Meersman, R., Tari, Z. (eds.) OTM 2007, Part I. LNCS, vol. 4803, pp. 131–149. Springer, Heidelberg (2007)

[5] Bider, I.: State-oriented Business Process Modeling: Principles, Theory and Practice. PhD thesis, Royal Institute of Technology, Stockholm (2002)

[6] Swenson, K.D. (ed.): Mastering the Unpredictable: How Adaptive Case Management Will Revolutionize the Way That Knowledge Workers Get Things Done. Meghan-Kiffer Press, Tampa (2010)

[7] iPB Reference Manual. On-line documentation, `http://docs.ibissoft.se/node/3`

[8] Bider, I., Johannesson, P., Perjons, E.: In Search of the Holy Grail: Integrating social software with BPM. Experience Report. In: Bider, I., Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Ukor, R. (eds.) BPMDS 2010 and EMMSAD 2010. LNBIP, vol. 50, pp. 1–13. Springer, Heidelberg (2010)

[9] Jin, X., Wei, H.: Scenario-based comparison and evaluation: Issues of current business process modeling languages. Proc. Inst. Mech. Eng. Pt. B: J. Eng. Manuf. 220(9), 1527–1538 (2006)