

To be published in BPMDS 2010 Proceedings

In Search of the Holy Grail: Integrating social software with BPM Experience report

Ilia Bider^{1,2} Paul Johannesson¹, Erik Perjons¹

¹DSV, Stockholm University, Stockholm, Forum 100, SE-16440 Kista, Sweden

²IbisSoft AB, Stockholm, Box 19567, SE-10432 Stockholm, Sweden
ilia@ibissoft.se, <pajo.perjons>@dsv.su.se

Abstract. The paper is devoted to finding a view on business processes that helps to introduce into business process support systems a notion of shared spaces widely used in social software. The paper presents and analyses the experience of the authors from a number of development projects aimed at building business process support systems. The authors define a role that shared spaces can play in business process support and set some requirements on the shared space structure based on this role. They then analyze their projects in order to show how these requirements can be met and describe what practical results have been achieved in each project.

Keywords: business process, information logistics, social software, state-oriented

1. Introduction

One of today's trends is a growing use of social software, e.g. Facebook, Twitter and Flickr, in private life. A new generation is growing up that is accustomed to communicate with each other through social software. Through this generation, this new way of communication is quickly spreading to business life. Business-oriented sites, such as LinkedIn, are widely used for informal business networks, personal marketing and sales. The ideas built into social software has begun to affect the design of business-oriented software systems, including Business Process (BP) support systems, which is reflected in several new directions in contemporary IS research [1].

Social software is based on the idea of common spaces shared by many individuals. This kind of software is used mainly for ad-hoc communication. Business process support is a system that helps process participants to drive their processes in a structured way towards specific "operational" goals. The question is how to "marry" these two on the surface different worlds in order to attain the benefits of social software when running business processes using a BP support system. The majority

of today's BP support systems are built upon the workflow view on BP. It is difficult to see how to add the idea of shared spaces to this view in a natural way. Another view on BP is needed for merging social software and business process support.

This paper presents an experience report that describes our efforts in finding a view on BP that allows integration of the idea of shared spaces from ad-hoc social communication with the goal-orientedness of business processes. Our underlying theory in this search was (and still is) the state-oriented view on business processes [2]. In this theory, a business process instance is defined as a trajectory in a state space, the driving forth of movement in which being people completing various activities (tasks). Activities are planned based on the position in the state space and sometimes on the process history.

Our initial practical experience has shown that a direct implementation of the ideas from [2] in a BP support system has a number of drawbacks. Consequently, some amendments to this view have been made to design BP support systems that both implement the idea of shared spaces and are convenient for end-users. The amended view has some similarity with the workflow view as it represents a process (type) as a number of boxes placed one after another (with the possibility to put some of the boxes upon others). The semantics behind the boxes is, however, different. Each box represents a subspace of the process state-space. Positioning of the boxes reflects the requirements on the instance trajectories, e.g., a progress in some subspaces is required before starting movement in some other subspaces.

Our experience report is structured in the following way. In section 2, we explain our view on a role of shared spaces in a system that supports BPs. In section 3, we formulate requirements on shared spaces for BP support. In section 4, we describe our initial experience of introducing shared spaces in BP support systems and explain why the initial plan has not been as successful as we hoped. In section 5, we describe how to build a BP support system based on the amended state-oriented view and discuss the advantages that it brings. Section 6 contains concluding remarks and plans for the future.

2. A role of shared spaces in BP support systems

Most of the young students whom we asked the question on what is the best thing with social software gave a straightforward answer: communication possibilities. You publish something, e.g., a photo album, once and can then make it available to as many people as you like and when you like it by inviting them to visit your space. The answer comes as no surprise, as communication is what social software is designed for. If we turn our attention to BP support systems, communication is not a primary objective here. The focus is on reaching the goal set for a process instance with as little communication as possible in order to be efficient. To introduce shared spaces in BP support, we first need to find their possible role in running business processes.

Roughly, there are two types of communication in the frame of a process instance, communicating the assignment of tasks and communicating information needed for completing the tasks. Communicating assignments is not exceptionally difficult, and

most BP support systems handle it satisfactory. Communicating information needed for the task execution represents a bigger problem, as it is not always possible to know beforehand how much information may be needed for completing a task in a particular process instance.

In this paper, we focus on the second type of communication – communicating information needed for the task execution. In fact, on a more abstract level, the goal here is not communication, but providing information. In analogy with the physical world, we can employ the concept of information logistics when dealing with this second type of communication in the frame of a process instance.

The term *information logistics* is relatively new. According to Wikipedia (which does not contradict with other sources on the matter), information logistics means: “providing the right information to the right recipients at the right time and place”. We do not fully agree with this definition, as, in our view, it implies “moving information to recipients”. This is in collision with the Requirement Engineering rule of maximum separation between the problem and solution domains. A better definition of information logistics for our purposes would sound like “bringing together information and people (or other type of agents, e.g. machines) who should process this information in the frame of a business process (instance)”. This is a more neutral definition, as it allows various logistics schemes, such as:

- Moving information to people
- Moving people to information
- Or any combination of the first two

Let us consider an analogy with production processes. In a production process, the term logistic means “bringing together physical objects and people (or other agents) who will complete some operations on them”. In production, both a scheme when objects are moved to people, a.k.a. conveyor belt logistics, and a scheme when people are moved to objects, a.k.a., construction site logistics, are widely used, see Fig. 1.



Fig. 1. Two types of logistics in production processes: to the left - conveyor belt logistics, to the right - construction site logistic

As far as business processes are concerned, the predominant way of arranging information logistics was, and still is, the conveyor belt scheme. This is quite understandable, because:

- Moving information, e.g. via mail, has been much cheaper than moving people
- Arranging people movement in an office would be a challenging task, see Fig. 2.

Introducing a BP support system moves us from the physical world to the virtual one. Movements in the virtual world do not cost much, and they are easy to arrange. For example, it is easy to move between different on-line bookstores, and people do not run into one another while doing so. Therefore, the costs and difficulties of arranging people movements are no more reasons to prefer the conveyor belt information logistics rather than the construction site one when designing business process support.

In addition, let us look at the production analogy more attentively. The conveyor belt logistics is extremely efficient for producing the same kind of goods over and over again, but nobody sets a conveyor belt if one needs to produce a personal car, a bus, and a lorry at random. This is exactly the kind of situations for many business processes; one instance can be as different from another as a personal car from a lorry. Thus, the efficiency of the conveyor belt in production may not be easily translated to its efficiency for business processes. At least, there is no logical reason for such an assumption.



Fig. 2. Moving people to information in a physical world is a challenging task

When there are substantial differences between the instances of a business process, it is difficult to decide what and how much information needs to be sent to a person completing a certain task. Choosing a construction site logistic here has an advantage. If we move a worker to a certain place inside a construction site, he oversees not only this local place, but also everything adjacent to it, and can use this information, if necessary, when completing his/her task without being explicitly told to do so. The same is true for business processes. When you send just one document to a person, this is all he/she gets. If you send a person to work on a certain document placed in some corner of a desk (see Fig. 2), he/she can access not only this document, but also other documents in this corner, or on the whole desk.

The above deliberations (more on this see [3]) lead us to the conclusion that the construction site information logistics can be preferable for business processes that are supported by a software system. A shared space in such a system plays a role of a construction site: it holds all information that is relevant to a process instance, e.g., document received and sent, information on tasks planned and completed, reports on results achieved when completing these tasks, etc. All this information is easily available each time a process participant is invited to visit this space and complete some task related to it.

3. Requirements on shared spaces for BP support

The functioning of a BP support system based on shared spaces can be described in the following way. Let us assume that the system has a capacity of creating any number of shared spaces, then:

- When a new process instance/case starts, a new shared space is created. It gets a unique name, an owner (responsible for the case), and possibly, a case team.
- When the process instance reaches its operational goal, the shared space is closed (sealed), but remains accessible for reading (a case goes to the archive).
- A person who is assigned a task in the frame of the process case “goes” to the shared space of the case in order to get the information he needs for completing the task and reports the results achieved in the same space.
- A shared space can be public, private, or restricted. If public, any worker can pop up in the “unlocked space” to see what is going on, and leave some traces of his visit, e.g. personal comments. If private, only the owner and members of the case team have access to the space. Restricted means the access is controlled by some rules specifying who can enter and who can see and do what in a restricted shared space. The rules are based on the position a person holds inside the organization, or/and the role he/she plays in the particular process case.

To make the above scheme work, we also need to provide a mechanism for issuing invitations to process participants to visit shared spaces and complete tasks in the frame of respective process instances/cases. There can be different solutions for this problem. The ones that we used in our practice are described in the next sections.

In the system described above, there is no information flow. A person is invited to visit a shared space and complete a task in it with the assumption that all information he/she needs is already there. In a normal business environment, a worker participates in many process instances and often in parallel. For the above scheme to work efficiently, he/she needs to understand the situation in a shared space he is visiting at a glance. This leads us to the requirement that each shared space should be highly structured, as nobody can work efficiently in unstructured shared spaces, e.g., as presented in Fig. 2. More important, shared spaces that belong to the same process type should be structured in the same way. The structure should facilitate easy understanding of in what state the process instance is and allow a person to quickly find all information related to the task at hands.

Summarizing the discussion of this section, the key to using the idea of shared spaces in BP support system is a proper structure of shared spaces.

4. Experience with a “static” structure of shared spaces

4.1 Short description

Our initial idea for structuring shared spaces in BP support systems was to directly follow the state-oriented approach from [2]. A number of BP support systems have been implemented based on this idea, among which a system called ProBis is the most

representative one [4,5]. In ProBis, a shared space consists of two parts, a static part that reflects the structure of the underlying state space, and a dynamic part – process plan and history.

A shared space is presented to the end-user as a form/window separated in several areas by using the tab dialogues technique, see Fig. 3. Some areas of the window are standard, i.e. independent from the type of the business process, while others are specific for each process type supported by the system. Standard areas comprise such attributes and links as:

- Name and informal description of a process instance
- Links to the owner, and, possibly, the process team
- Links to the relevant documents, created inside the organization and received from the outside

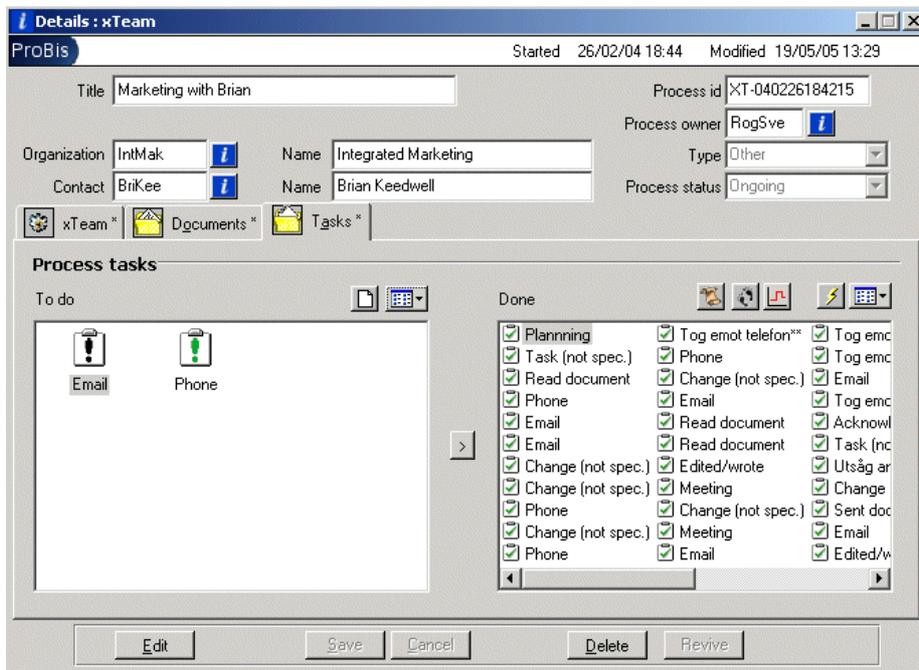


Fig. 3. A static structure of a shared space employed in BP support system ProBis (see [4]).

The standard part of ProBis shared space includes also the task area (tab) that contains two lists, as in Fig. 3. The “to-do” list (to the left on Fig. 3) includes tasks planned for the given process instance; the “done” list (to the right on Fig.3) includes tasks completed in the frame of it. A planned task defines what and when something should be done in the frame of the process instance, as well as who should do it. In ProBis, the process plan serves as a mechanism for issuing “invitations” to attend a particular shared space. All “invitations” from all process instances are shown in an end-user’s personal calendar, see Fig.4. From the calendar, the user can go to any shared space to which he was invited in order to inspect, change, or execute a task planned for him/her.

In case of a change in the user's planned tasks, e.g., when a new task is added to some process instance and assigned to him/her, a pop-up window appears to inform the user about the change. If the user is not on-line, an email message is sent advising him/her to log in and view the changes.

The "done" list shows all events that already happened in the frame of the given process instance, independently of whether they appear there as results of planned tasks execution or as ad-hoc changes in the process state. An event (completed task) shows the date and time of when it happened, participants of the event, comments on it, etc.

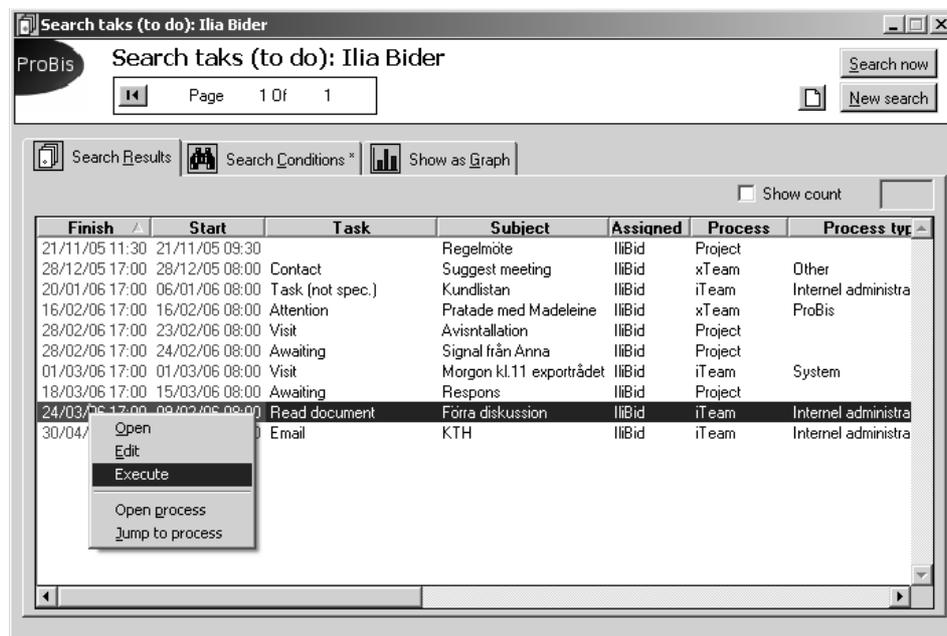


Fig. 4. A person's calendar serves as a mechanism for inviting him/her to visit shared spaces

Process participants work with the shared spaces in ProBis in the following manner. A participant comes to a shared space because a task has been planned for him/her in this space, or in the ad-hoc manner while browsing through the list of existing shared spaces (i.e. opened process instances/cases). When in the space, he/she can decide to make changes in it by changing the values of various fields, attaching new documents or persons to the shared space, etc. Any change in the shared space results in adding an event to the "done" list of the tasks tab (Fig. 3). If the change is due to the execution of some planned task, the event represents a report on the execution, otherwise the event represents some ad-hoc activity. In the simplest case, a process participant just moves his/her planned task from the "to do" list to the "done" list and presses the save button. He/she may also choose to write a report or attach a document to the event.

When changing a shared space, a participant can make changes in its plan (to-do list) by adding new tasks, or augmenting or deleting the existing ones. When inserting

a new task he/she can plan it for him-/herself or to any other person. The latter serves as an invitation for this person to visit the shared space.

The above scheme provides several layers of information to a person when he visits a shared space. For example, if he visits the space to complete a planned task the following information is available to him/her:

- Task description, which includes a name of the task and its parameters. The name informs what action to complete, like contact somebody, read or write a document, etc. The parameters provide additional information, like whom to contact (link to a person), what document to read (link to a document), textual description with additional instructions, who planned the task, etc.
- Reference to the event from the “done” list that has caused this task to appear in the to-do list.
- All information already in the shared space, values of various fields, documents attached to the process instance, etc.
- “Done” list that functions as a full chronicle (log) on what has happened in the frame of the instance.
- “To do” list that provides information on what is going to happen and helps to avoid double planning.
- Full historical information about the process instance. A user visiting the shared space can see what it looked like before or after any event registered in the “done” list, or browse through the past states of the shared space one by one in the forward or backward direction.

The user visiting the shared space decides for him/herself how much information he/she needs for completing the task at hands. He/she can satisfy him-/herself with the task description, or scrutinize the whole case, including full history.

4.2 Lessons learned

Based on our experience with ProBis, one thing is certain: a system of this kind provides a very efficient way of communication in the frame of business process instances. It is especially useful for:

- loosely structured processes, i.e. the processes for which there are no predefined ways for handling each case
- driven by a professional team that knows how to use the system quite well

There are, however, two main drawbacks with the approach when using it for more structured process or/processes that involve occasional users:

1. The dynamic aspect of business processes is poorly visualized. One needs to go through the done-list and browse through the history to get an understanding of how a given process instance is developing in time.
2. To use the system puts some requirements on the user, as he/she needs to understand the general ideas built in the system and get some training. This means that the system is not very friendly for newcomers and casual users. Planning as a way of issuing invitations causes the major problem here, as it is considered to be

counter- intuitive. Detailed planning is not as widespread in business life as one can imagine.

We found that these two drawbacks above considerably hamper the possibility of utilization of systems like ProBis for structured processes with many occasional or untrained users. This is especially true in the current business environment where end-users more or less refuse to read manuals and demand the system being so intuitive that one can fully understand it by using try-and-error techniques. This observation has led us to rethinking the whole concept and designing a new way of structuring shared spaces that is better suited to the purposes mentioned above.

5. Experience with a “dynamic” structure of shared spaces

5.1 Short description

Our latest system is called iPB [6]. In contrast to ProBis, it is not a ready-made process support system, but a tool (more exactly a web service) for developing such systems. In an iPB-based system, shared spaces are structured according to the process map designed for a particular process type. A process map in iPB is a drawing that consists of boxes placed in some order, see Fig. 5. Each box represents a step inside the process, the name of the step appearing inside the box (no lines or connectors between the boxes). A textual description is attached to each step that explains the work to be done in this step.

Each process instance gets its own copy of the map that serves as a table of contents for its shared space. The map is used for multiple purposes: as an overview of the case, guidelines for handling the case, and a menu for navigating inside the shared space, see Fig. 6. The user navigates through the shared space by clicking on the boxes of the steps with which he/she wants to work. Not all boxes are clickable at the beginning, those that are grayed require that one or several previous steps are dealt with first, see Fig. 6.

A click on a step box redirects the end-user to a web form that assists him in completing the step, see Fig.7. The form contains text fields, option menus and radio-buttons to make choices, checkboxes, as well as more complex fields. The form may also include “static” texts that explain what should be done before one can fill some fields.

The progress in filling the step forms is reflected in the map attached to the shared space via steps coloring. A gray box means that the step form has not been filled and cannot be filled for the moment. A white box means that the step form is empty but can be filled. A step with a half-filled form gets the green color, and additional information about when the work on it has been started, and who started it. A step with a fully filled form gets the blue color, and additional information about the finish date.

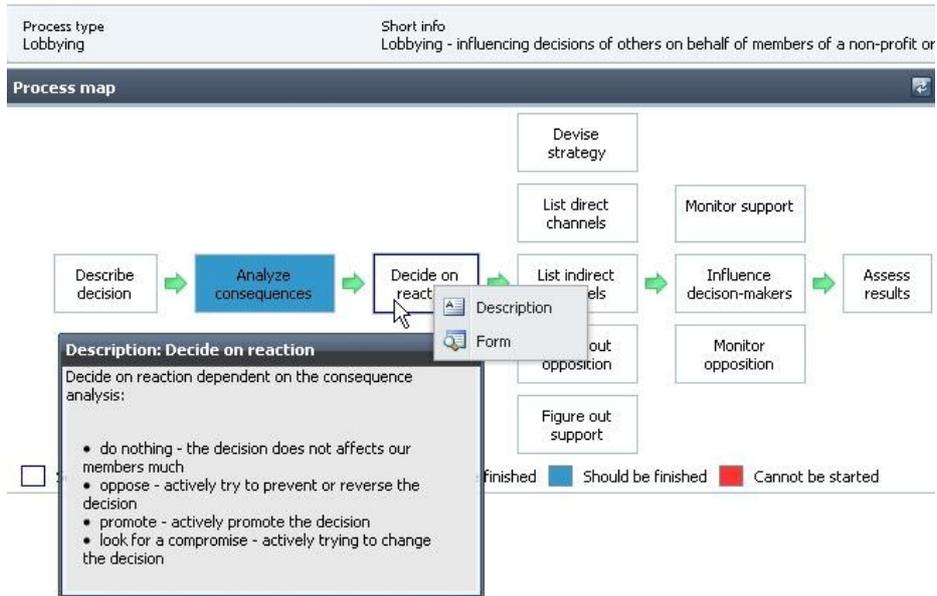


Fig. 5. A process map in iPB



Fig 6. The map used for structuring an instance shared space

The main way of inviting a person to visit a particular shared space in iPB is by assigning him/her to become an owner/co-owner of some step. Such an assignment results in an email message delivered to this person, and the process to appear in his/her list of “My processes”. When visiting a process shared space, a person can see directly on the map what step(s) are assigned to him. Optionally, the same scheme of planning tasks as described in the previous section can be used.

Fig 7. A step form for the first step from Fig. 6.

5.2 Underlying theoretical model

From the theoretical point of view, the approach described above represents a modification of our state-oriented view on business processes [2]. The basic ideas behind this modification consist of the following:

- The total process state-space is divided into a number of subspaces called process steps. The steps are graphically represented to the end-users as boxes. Subspaces may or may not intersect. The structure of a step subspace is represented to the end-users as a form to fill, see for example Fig. 7. Intersecting subspaces means that web forms attached to different steps may contain the same field(s). Usually, in this case, the intersecting fields can be changed only in one form; they are made read-only in the second one.
- The steps are ordered in a two-dimensional matrix that defines a recommended strategy of movement in the state space. The movement starts in the top leftmost subspace and continues in the top down left to right order. This matrix does not prohibit any other way of movement through the subspaces. For example, it allows parallel movements in several subspaces. The matrix is presented to the end-users in the form of a process map, see, Fig. 5, and 6.
- The restrictions on movement through the subspaces are defined with the help of business rules. Such a rule, for example, may require that movement in one subspace should be finished before the movement in another one can be started. Business rules are represented to the end users via gray boxes – steps that cannot be handled yet. Clicking on a gray box results in a message that explains why the box is gray, e.g. that some other box should be started first.

5.3 Lessons learned

Our experience of introducing iPB-based systems into operational practice shows that end-users, even new ones, have no major problems in understanding the structure of shared spaces, and they learn to navigate in them very quickly. Users appreciate the idea of the multipurpose map that gives them an instant overview of the case, and simultaneously serves as a tool for navigating in the shared space. Our practical experience so far gives us a hope that we, at last, found a right approach to structuring shared space in BP support systems.

Conclusion

We started with the idea of introducing shared space technique from the social software into BP support systems. In the first part of this paper, we analyzed the role that shared spaces, very well known from the groupware research (see, for, example, [7]), could play in BP support using the concept of information logistics. Based on this analysis, we concluded that a shared space could serve as a kind of a “construction site” that contains all information related to a given process instance/case. For the idea to work in practice, the shared spaces need to be properly and uniformly structured. This structure should reflect the peculiarities of the given process type(s) that the system is aimed to support.

In the second part of this paper, we described two cases of realization of the idea in real BP support systems. The first case represents a system with rich functionality, which, however, lacks proper visualization for structured processes, and requires extensive training before it can be used in operational practice. The second system is highly visual and easy to learn, though it, for the moment, lacks some functionality that can be found in the first system, e.g. full history.

Both cases are based on the state-oriented view on BP from [2]. In the first case, the original idea has been used as is. The second case exploited a modification of the original idea that consists of splitting the total process state space into subspaces, and introducing some order between them. The order is introduced in two ways, via relative positioning (recommended order), and via business rules (hard restrictions). The modification makes it possible to represent the process space as a map that is somewhat similar to the traditional workflow, though it has different semantics. This map makes it possible to achieve much better visualization of shared spaces of structured processes to the end-user than when using the original approach.

Our experience of introducing BP support systems into operational practice shows that the end-users appreciate the support provided by the systems built on the modified state-oriented view. Thus, we conclude that the idea is promising but needs to be developed further. Our future plans include developing the split state space model in more details theoretically as well as continuing to further develop software based on it.

The theoretical part concerns relationships between subspaces, which will enrich iPB with a more elaborated system of business rules. The practical part consists of

moving the rich functionality built in ProBis to iPB to satisfy the future demands from the more experienced users.

Acknowledgements. This paper would have never been written without considerable efforts of the team of developers who have designed and implemented both ProBis, and iPB. We are especially thankful to Tomas Andersson, Alexander Durnovo, Alexey Striy and Rogier Svensson.

References

- [1] Nurcan S., Schmidt R.. Introduction to the First International Workshop on Business Process Management and Social Software (BPMS2 2008). *Business Process Management Workshops 2008*: 647-648, Springer 2009
- [2] Khomyakov M., and Bider, I. Achieving Workflow Flexibility through Taming the Chaos. *OOIS 2000 - 6th international conference on object oriented information systems*: 85-92. Springer, 2000.
- [3] Bider I. *New Logistics for Administrative/Business Processes*. White paper, IbisSoft, 2006, p.14: <http://www.ibissoft.se/whitepapers/newlogistics.pdf>
- [4] Andersson T., Bider I., Svensson R. Alignig people to business processes. Experience report. *Software Process: Improvement and Practice (SPIP)*, Willey, V10(4), 2005, pp. 403 - 413.
- [5] Bider I., Striy A. Controlling business process instance flexibility via rules of planning. *International Journal of Business Process Integration and Management (IJBPIIM)*, Inderscience. VOL 3(1), 2008, pp. 15-25.
- [6] *iPB Reference Manual 8on-line documentation*). IbisSoft, 2009: <http://docs.ibissoft.se/node/3>.
- [7] H. Takemura und F. Kishino, "Cooperative work environment using virtual workspace," *Proceedings of the 1992 ACM conference on Computer-supported cooperative work*, ACM New York, NY, USA, 1992, p. 226-232.