Key challenges for Enabling Agile BPM with Social Software

	Affiliation	E-Mail	Address	Telephone
Giorgio Bruno	Dip. Automatica e Informatica, Politecnico di Torino	giorgio.bruno@polito.it	Corso Duca degli Abruzzi 24, 10129 Torino, Italy	+39 011 5647003
Frank Dengler	Institut AIFB, Karlsruhe Institute of Technology	frank.dengler@kit.edu	KIT-Campus Süd Geb. 11.40 76128 Karlsruhe Germany	+49 (0) 721 608 - 6584
Ben Jennings	University College London	b.jennings@cs.ucl.ac.uk	44 Hartswood Avenue, Reigate, Surrey UK	+447595037293
Rania Khalaf	IBM TJ Watson Research Center	rkhalaf@us.ibm.com	1 Rogers St, Cambridge MA 02142	+1 617 693 5084
Selmin Nurcan	* University Paris 1 Panthéon Sorbonne ** IAE de Paris - Sorbonne Graduate Business School	nurcan@univ-paris1.fr	21, rue Broca 75005 Paris France	+33 - 1 53 55 27 13
Michael Prilla	University of Bochum, Institute for Applied Work Science, Chair of Information and Technology Management	Prilla@iaw.ruhr-uni-bochum.de	Universitaetsstr. 150, 44780 Bochum	+49 (0) 234 32 27735
Marcello Sarini	Department of Psychology – University of Milano-Bicocca	marcello.sarini@ unimib.it	Piazza Ateneo Nuovo, 1, Milano – Italy	+39 (0)2 64483746
Rainer Schmidt	Aalen University	Rainer.Schmidt@htw-aalen.de	Anton-Huber-Str. 25 73430 Aalen Germany	+49 (0) 7361 576 - 4241
Rito Silva	INESC-ID/IST/Technical University of Lisbon	Rito.Silva@ist.utl.pt	Rua Alves Redol 9 1000-029 Lisboa Portugal	+351 213100287

Abstract

Business Process Management is called agile when it is able to react quickly and adequately to internal and external events. Agile Business Process Management requires putting the lifecycle of business processes on a new paradigm. It is advocated in this paper that social software allows us satisfying the key requirements for enabling agile BPM by applying the four features of social software: weak ties, social production, egalitarianism and mutual service provision. Organisational and semantic integration and responsiveness (of the business processes engineering, execution and management activities) have been identified as the main requirements for implementing an agile BPM lifecycle. Social software may be used in the BPM lifecycle in several manners and using numerous approaches. This

paper presents seven among them and then analyses the “support” effects between those approaches and the underlying social software features, and the three requirements for Agile BPM. .

Contents

1 Introduction

Today, there is a growing awareness of shortcomings of the classical BPM approaches. Based on observations made in practice, issues such as the model-reality divide [1] have been identified. It describes the divide between abstract process models and the processes executed in practice.

Recent work has revealed a number of viewpoints or perspectives which exerted great influence on the design of notations and languages for business processes. These viewpoints propose different ways of organising the basic constituents of cooperative business environments, which are the operational business activities, the coordination activities and the common field of work [2]. Operational activities are units of work meant to produce some changes in the underlying common field of work which is a repository of artefacts (business data and documents). Coordination activities are responsible for scheduling the operational business activities with respect to the business logic.

Business processes have often been roughly classified into two categories depending on their nature. The first concerns well-defined and often repetitive processes having important coordination and automation needs. The second category concerns loosely structured processes (also known more recently as knowledge-intensive). The essential preoccupation with the latter is the information and knowledge sharing between the actors implied in the processes more than the coordination of their activities. Business processes of this category require more flexibility. For many organisations, well-defined and knowledge-intensive processes coexist and must be handled in the final business model [3][4].

As studied more recently during the BPMDS'2010 workshop [5], various modeling languages and notations differ by which and whose perspective they take on a business process. In theory as well as in practice, the workflow/control-flow perspective, also well known as activity-driven, was dominant for a long time. For highly structured processes, the focus on activity-driven perspective satisfied the needs of BPM practice. In this context, the business process execution engine is like a master distributing the work among the participants according to a rigid control flow. A business process involves a number of participants denoted by their roles. Activities and control flow elements are the major building blocks for well-defined business processes while swim lanes are the major structuring mechanisms. Well-known notations and languages, such as BPMN [6] and BPEL [7], support the centralised (activity-driven and scheduler-oriented) viewpoint.

However, as the BP domain expanded to the less structured flexible processes in flexible environments, the one-sidedness of the modeling techniques and systems based on the dominant perspective became apparent [5]. It seems clear today that the knowledge-intensive business processes may not be represented in terms of rigid control flows and requires new perspectives (goal, state, context, actor, ...) to understand and to share their complexity among stakeholders being actors in the said business processes or designers.

The idea of integration of structured and unstructured processes is not new. For example, Sadiq et al. introduced in [8] the so called Pocket of Flexibility approach which allows parts of the business process being unspecified during build-time. Also Pesic describes in her PhD thesis [9] how procedural and constraint based processes can be combined. The integration of pockets of creativity in a well-structured business process can be seen as an integration of small groups carrying out loosely structured, knowledge intensive, social processes in (or beside) a larger group processing a globally well-structured business process as studied earlier in [3] and more recently in [10].

The use of group-oriented models (instead or in addition to control flows), in order to represent stages of the cooperative work (synchronous work which could not be represented using usual workflow models) has been proposed in previous research works [4], [11], [12].

Further issues identified in [13] are the accelerated pace of changes, the spreading of context information of all kind of business processes across different systems, such as e-mail, documents etc., and a growing demand for supporting user generated "instant-BPM", quickly created process solutions. Such a BPM capable to react quickly to external and internal events shall be named agile BPM. Agile BPM is needed to implement the vision of an agile enterprise [14] that is capable to rapidly adapt to changing business challenges and opportunities. Agile BPM is the basis for dealing with adhocracy [15] or "pockets of adhocracy" in other kinds of organisations and to foster the creative potential. Furthermore, agile BPM is necessary to cope with the numerous evolution requirements in today's organisations. Its characteristics become clear if compared to standard BPM and its lifecycle.

The standard BPM lifecycle often consists of a number of phases such as design, deployment, operation and evaluation [16] even if literature provides research works [17][18] that aim to enhance this lifecycle to make it more flexible (or more precisely, less rigid). During the BPM lifecycle, interactions have to take place both between the stakeholders of the process. These interactions may span multiple phases of the lifecycle. Standard BPM is based on the understanding that these interactions follow ordered steps and use more or less predefined flows of information. The steps and procedures, the flow of information and the role of the participants are rigidly defined and allow only few modifications.

However, in this way the capability to cope with external events is restricted to those which are "foreseen" in the structure of the business process lifecycle. Furthermore, the pre-defined roles of the stakeholders also impede information flows not fore-seen by the business process life cycle. Thus, stakeholders may be not able to provide the necessary information, because they are not requested or allowed to do it. Another drawback is the a-priori quality assurance approach. Changes have to undergo detailed procedures to be regarded of good quality. However, these are only formal steps, not really assuring the quality of the outcome.

Therefore, to cope with the issues identified above and to create an agile BPM, it does not help to create yet another BPM lifecycle with further phases inserted or other manipulations. Also simply loosening the "grip" of the BPM lifecycle does not help either. Actually, agile BPM not only requires minor changes to the BPM lifecycle, but a paradigmatic change. This new paradigm for BPM shall at the same time provide (i) an appropriate flow of information (more precisely an

effective knowledge sharing) and (ii) engineering / execution / management processes to handle the business processes efficiently, and assure that a high quality of business process modeling and execution is performed. It is advocated that such a change can be achieved by using social software.

The basic advantages of using social software in the engineering / execution / management of business processes have already been identified in [19]. Social software is a paradigm for software systems providing four features: weak ties, social production, egalitarianism and mutual service provisioning. It allows the spontaneous creation of contacts between non-predetermined individuals, called weak ties. Social software also supports social production as defined by Benkler [20] or Tapscott [21]. Social production creates artefacts, by combining the inputs from independent contributors without predetermining the way of performing. Social software realises egalitarianism by merging the roles of contributors and consumers and introducing a culture of trust instead of formal access procedures. Mutual service provisioning is another feature of social software. By combining mutually provided services, a new service is rendered.

The contribution of this paper is to show how social software can be used to cope with some key challenges of an agile BPM lifecycle. It is not the goal, to deliver a complete agile lifecycle. However, it shall be shown, how social software helps improving the organisational and the semantic integration and the responsiveness of the BPM lifecycle. It is not the goal either to focus all kinds of business processes but more specifically the knowledge intensive and collaborative ones requiring creativity, flexibility, agility, and responsiveness instead of tightly fixed coordination rules and procedure driven standardisation.

To achieve the goal of understanding how social software can be used to deliver elements for an agile BPM lifecycle, the prerequisites for such an agile BPM lifecycle are analysed in detail. Then, the four features of social software: weak ties, social production, egalitarianism and mutual service provisioning are presented. Based on this introduction, approaches for supporting organisational and semantic integration and responsiveness by social software are introduced. All approaches discussed in this paper, have been introduced and presented during the BPMS2'09 (BPM and Social Software) [22] workshop in conjunction with the BPM'09 Conference and their leading authors are involved in the present work.

2 Prerequisites for an Agile BPM Lifecycle

An agile BPM lifecycle is a BPM lifecycle capable to support the vision of an agile enterprise. Therefore it should be able to react quickly to external and internal events. However, present BPM lifecycle approaches are based on tayloristic thinking. That means that BPM lifecycle has to be divided up into small procedures which are often exclusively assigned to a single stakeholder. The rights and capabilities of the stakeholders to contribute are therefore tailored in order to support the execution of the assigned operations. On the other hand they restrict the contributions of the stakeholder to just those ones, pre-defined by the BPM lifecycle. This may cause, that a multitude of communication bottlenecks created [23] such as between process designers and stakeholders or implementers and management.

In consequence of tayloristic thinking, the organization of procedures is also very rigid. There are nearly no possibilities to change the order of procedures. The rigidity in role assignment and

ordering of procedures has its roots in an a-priority quality assurance approach. The predefining of roles and ordering of procedures shall assure that the business process is appropriately managed during its life cycle.

Another source for identifying the prerequisites is the analysis done in [23]. There, the failure to use a language understandable to all process stakeholders has been identified as risk for business process management in general. Because it may delay the reaction to external events, this risk is of particular importance for an agile BPM lifecycle. Furthermore, the lack of a strategic view is also regarded as a risk for business process management.

To become agile, the BPM lifecycle has to achieve integration and responsiveness, as shown in Figure 1: a) organisational integration has to overcome organisational barriers which exclude stakeholders from submitting their requirements; b) semantic integration has to break barriers impeding the appropriate fusing of the requirements; c) responsiveness has to break barriers created by engineering & execution & management processes[1] dealing with business processes through their lifecycle which do not appropriately support information and knowledge sharing within the BPM lifecycle.

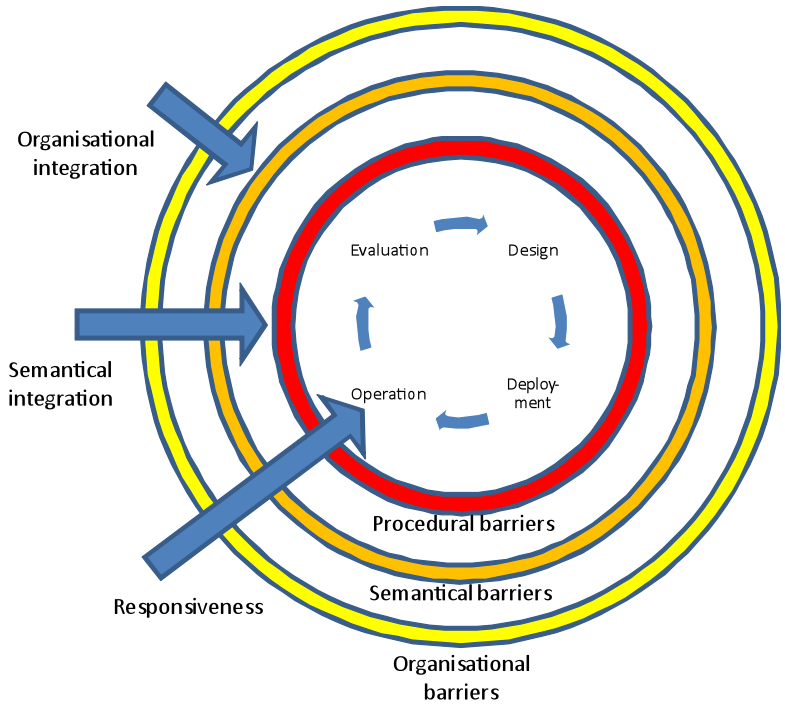

Figure 1: Prerequisites for agile BPM Lifecycles

2.1 Organisational Integration

Agile BPM requires the organisational integration of all stakeholders. A priori, it is not possible to identify all valuable contributions and their contributors. Therefore, every stakeholder has to

[1] we will call them, in a generic way, *BPM processes* in the following

be capable of submitting his requirements to the BPM lifecycle. This approach is supported by research, which has shown that in many areas combining a multitude of opinions leads to the best results [24]. In standard BPM approaches, an a priori decision about contributions is made by pre-defined groups. The group membership decides the extent to which an individual may contribute to the BPM lifecycle. Furthermore, the weight of the contributions differs highly from group to group, e.g. consultants get much more influence than ordinary users and the breadth and the quality of requirements elicitation is diminished.

Therefore, in agile BPM organisational integration has to overcome elitist and hierarchic views on the world. It has also to assure that stakeholders, not regarded as capable in standard approaches, are integrated into the requirements elicitation process and thus able to submit their requirements to the BPM lifecycle. The organisational integration has to break both explicit and implicit forms of exclusions, such as the exclusion of stakeholders not belonging to certain organisational units or the exclusion of people not trained in formal process modeling. Thus, they are disabled from contributing their requirements. One has to consider that the means of exclusion may be very subtle, e.g. not providing proper means to submit the requirements.

2.2 Semantic Integration

To be able to react quickly to change requests, semantic frictions should not slow down the BPM lifecycle. Semantic frictions such as homonyms and synonyms create misunderstandings which may prolong the process definition. Even worse, if they are detected in later phases, they may require a huge effort to be removed and thus degrade the performance of the BPM processes. Semantic frictions may be imposed by a certain world view or terminology, either from an external consultant or from a designated member or group within the team. They also cause a lack of depth in requirements elicitation, because homonyms and synonyms impede a deeper analysis of requirements. Although the requirements are identified, they are not properly fused into a consistent overall specification. To avoid these problems, agile BPM needs semantic integration to create a common understanding of the terms and their relations used in the BPM lifecycle among stakeholders.

2.3 Responsiveness of engineering, execution and management processes

To be able to react quickly and properly to internal and external events, an agile BPM requires responsiveness. Responsiveness is the capability of the BPM lifecycle to adapt its structure, which means the flexibility of BPM processes and information flows in order to react to internal and external events. Standard BPM approaches have predefined control and information flows. They are often more a burden than an aid because they inappropriately restrict the control and information flow within the BPM lifecycle. Although some may allow the reaction to predefined events, the basic structure is still rigid and not able to react appropriately to unforeseen events. Furthermore, the possibility to react is restricted to certain points in the lifecycle but not available throughout the lifecycle. A common flaw of BPM lifecycles is that there is only a top-down but not a bottom-up flow of information in real-time. As a result, information available and semantically integrated with the BPM lifecycle is not usable.

3 Features of Social Software

As introduced above, social software is software that is built according to four features, which will be described in this section. Afterwards, the ways they can support the prerequisites for being able to provide an agile BPM, introduced in section 2, will be developed in section 4.

3.1 Weak Ties

Relationships that are based on hierarchy or team structure, called strong ties [25], are the core of enterprise organisation and are probably appropriate for well-defined and repetitive business processes. However, they are less able to provide new perspectives on problems. Therefore, so-called weak ties [25] are very important for enterprises to innovate. As argued in [26] [27] "*An unstructured process cannot be represented in terms of flow of tasks. [Then the proposed generic model] allows to represent it associated to a set of resources that it uses and produces and a set of participating roles. The key concept of unstructured processes is the information and knowledge sharing in the work group*". Weak ties are spontaneously established contacts which create new views on problems and allow combining competencies. They are contacts not imposed by management but by the individuals. Before the internet, weak ties happened to be created, e.g. in the smokers' corner. Social software supports the creation of weak ties by allowing the spontaneous creation of contacts between non-predetermined individuals by a search mechanism on the profiles of registered members. Social software sites such as LinkedIn or XING demonstrate these capabilities quite impressively.

3.2 Social Production

Social Production breaks with the paradigm of centralised a-priori planning of production. By this means it is possible to integrate unforeseen and innovative contributors and contributions. "*Many social and organisational factors play an important role in the working of any organisation ... a useful cooperative work model must capture much more than the steps of procedures… The concept of goal expresses an intention, this is what must be achieved*" [26]. In other words, it is not always possible or even appropriate to freeze the way of reaching it. Important results of social production are Wikipedia and the Linux operating system. Furthermore, social production is based on a posteriori approach for assuring quality of production. The collective evaluation by all participants aims to reach and to keep a high degree of quality. An important precondition for such a collective evaluation is the independence of the participants. Influences biasing the participants have to be avoided.

3.3 Egalitarianism

Social software realises egalitarianism by abolishing hierarchical structures, merging the roles of contributors and consumers and introducing a culture of trust. Social software relies highly on the idea of giving all participants the same rights to contribute. This is done with the intention of encouraging a maximum of contributors and of getting the best solution by fusing a high number of contributions. In the same way, all participants have the right to contribute; they also have the duty to contribute. It is no longer possible to delegate tasks which the participant could do themselves.

3.4 Mutual Service Provisioning

Social software changes the cooperation model from a client-server model to a model based on the exchange of services. This thinking is closely related to Service Dominant Logic (SD-Logic) [28], where a "service is defined as the application of specialised competences (knowledge and skills) for the benefit of another entity, rather than the production of units of output" [28]. The co-creation of value and not the output of production should be the centre of interest. Thus service is regarded as a process of interaction with the customer and not as an interface to the customer. Value is co-created in a service process by a service provider and his customer, instead of producing goods and delivering it to the customer.

4 Social Software for Agile BPM

The transition from present BPM to an Agile BPM implies the change of essential paradigms of the BPM lifecycle such as the tayloristic approach used by present BPM lifecycles. Therefore, isolated changes to single phases of the BPM lifecycle cannot overcome the existing deficiencies. Instead, a fundamental rethinking of the design principles of the BPM lifecycle has to take place, affecting the lifecycle in a holistic manner.

To prepare this redesign of the BPM lifecycle, the requirements to satisfy for enabling agile BPM are analyzed. It will be shown on an abstract level, how social software is capable to fulfill these requirements. Then, approaches to fulfill these requirements will be presented.

4.1 Requirements to satisfy for enabling agile BPM

Organisational and semantic integration and responsiveness have been identified as requirements for an agile BPM lifecycle in Section 2. This section will analyse how the features of social software can help to fulfil these requirements.

Figure 2 sketches the framework which results from our analysis of "support" effects (+ in terms of system dynamics ([29], [30])) between the manners/approaches that authors of this paper developed in their own research using Social Software features (Section 3) and the three requirements we identified during our collaboration (Section 2) as prerequisites for agile BPM.

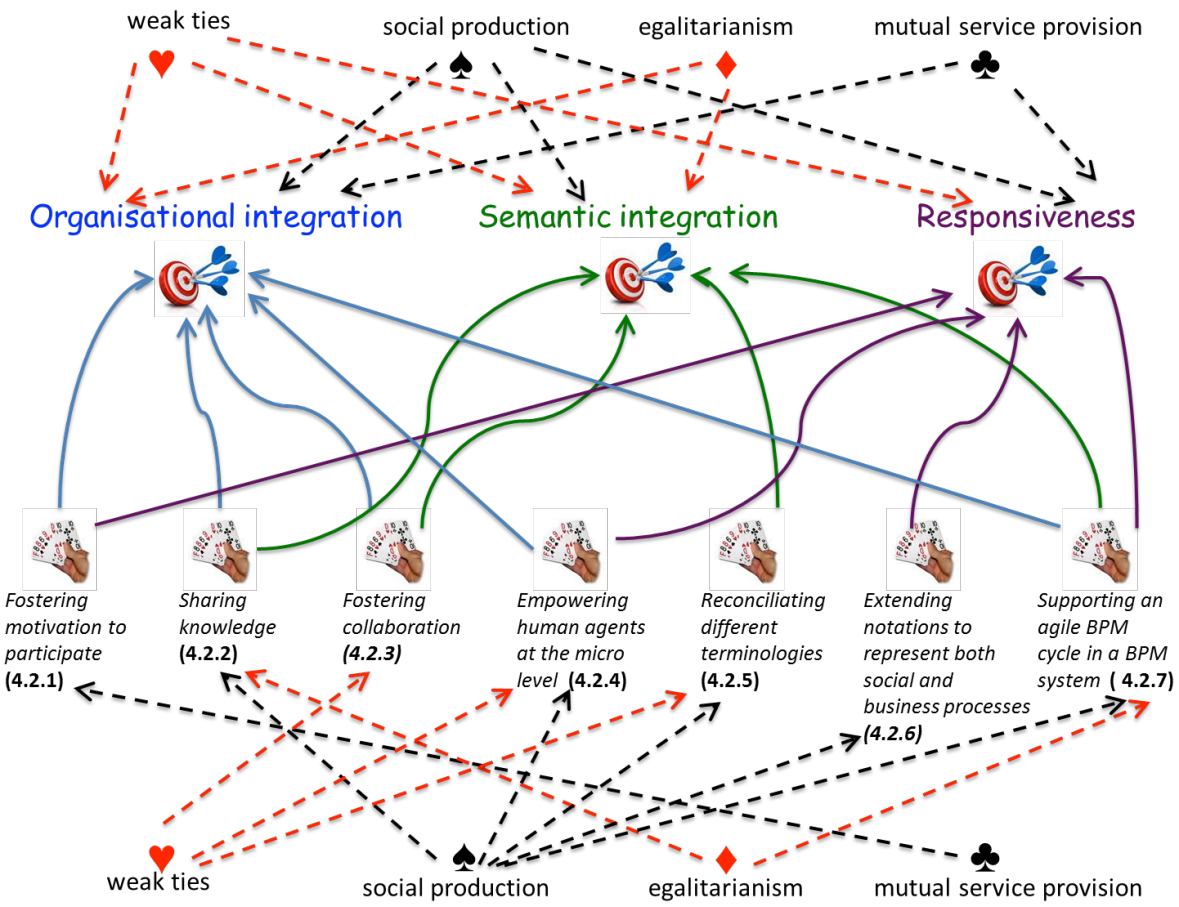

Figure 2: Mapping Social Software features and the approaches developed in this paper (Section 4.2) to agile BPM requirements

Based on this framework, concrete approaches will be presented in Section 4.2, showing how to achieve organisational and semantic integration and the responsiveness of the BPM lifecycle.

4.1.1 Organisational integration

Thus, we have to foster the motivation for actively dealing with business processes (cf. Section 4.2.1, enable people to acquire knowledge from and add their knowledge and requirements to these processes (Section 4.2.2), re-integrate the outcome of these tasks into processes and their execution (Sections 4.2.3 and 4.2.4) and provide a software (Section 4.2.7) that enables collaboration between process designers, end users who use them and software developers who create programs which extend existing workflow management systems and languages. Otherwise, people cannot contribute requirements to business processes because they are not provided with proper means to do so.

Organisational integration requirement may be satisfied taking into account the difficulties of involving stakeholders in providing their contributions during the phases of BPM life cycle. These difficulties can be dealt with by also considering socio-psychological aspects of human interactions. What is proposed is the use of social software in relation to these aspects, in particular by focusing on a social identity approach (cf. Section 4.2.1): people are motivated to participate if they feel part of the same group and develop a common identity.

Organisational integration can benefit significantly from the integration of stakeholders into the creation, adaptation and revision of business processes throughout the BPM lifecycle. For this integration to happen, these stakeholders must be aware of business processes (as well as BPM processes to some extent), the corresponding knowledge and the possibility of participation in the BPM lifecycle. Therefore, the knowledge of business processes and models depicting those processes should be available in organisational knowledge management systems, which are commonly used by most people in an organisation. This enables agile BPM, as business processes are published among staff and input as well as feedback on processes can be given spontaneously and contextually. The approach in Section 4.2.2 shows how process models and corresponding knowledge can be integrated into knowledge management systems with a social tagging mechanisms and how this integration corresponds with BPM systems.

To achieve organisational integration it is necessary to foster collaboration among stakeholders by allowing them to communicate using their own perspectives and languages. In particular, end users' perception of business processes occurs at the instance level, while executing business process instances. Moreover, in a dynamic and ever-changing organisational environment, end users react to unexpected and non-specified events by diverting the execution of business processes from their specifications. However, the rationale behind deviations is not communicated and cannot be used by business process experts to evolve the business process definition. The approach in Section 4.2.3 explains how the use of social software features at the instance level can be used to foster collaboration among stakeholders while preserving their perspectives and languages of the business process.

Weak ties support organisational integration by allowing people not designated by hierarchy to contribute to the BPM lifecycle. Thus they can provide valuable knowledge and information. Organisational integration is also supported by the egalitarianism found in social software. There are no pre-determined roles limiting the possibility of the individual to contribute to the BPM lifecycle. Social software uses an optimistic and a posteriori approach for determining possible contributors. There are no restrictions for submitting requirements; however the value of contributions is checked a posteriori by the community of stakeholders. This approach may lead to better results than the a priori one, when certain constraints are obeyed, as discussed in [24].

Mutual service providing feature (Section 3.4) of social software facilitates the motivation to participate (Section 4.2.1) and thus supports the organisational integration. Social production (Section 3.2) facilitates sharing knowledge (Section 4.2.2) and empowering human agents at the micro level (Section 4.2.4) and thus supports the organisational integration. Egalitarianism feature of social software (Section 3.3), facilitating knowledge sharing, participates also to the organisational integration. Finally the same applies to weak ties (Section 3.1) which participate to the organisational integration by empowering human agents at the micro level (Section 4.2.4).

4.1.2 Semantic integration

The semantic barriers can be overcome by fusing the worldviews of the stakeholders in the early phases of requirements elicitation. The involved stakeholders can have different goals, skills, values, languages, terminologies, points of views, motivations, knowledge of the processes. This can be compared with the definition of a common language of all stakeholders. To be successful, the creation of such a common language has to be done in a multitude of small steps, giving each stakeholder the ability to adapt easily to a common usage of terms. Social Production is important for this fusing of terms and knowledge in BPM. Using social software, the definition of terms and processes is transformed into social production, allowing all stakeholders to contribute. To this aim, techniques for facilitating the reconciliation of multiple terminologies and the representation of both social and business processes need to be considered.

An important means is the reconciliation of different terminologies as described in Section 4.2.5. The approach presented in Section 4.2.5 explains how Semantic MediaWiki can support the BPM lifecycle by creating and sharing a common used terminology.

Egalitarianism (Section 3.3) is an important feature for semantic integration (by the way of sharing knowledge, Section 4.2.2; and supporting the BPM cycle by a software system, Section 4.2.7), as only the equal handling of the views of all stakeholders will create a common understanding. Also social production (Section 3.2) is an important means of achieving semantic integration (by the means of sharing knowledge, Section 4.2.2; reconciliating different terminologies, Section 4.2.5, extending notations to represent both social and business processes, Section 4.2.6; and supporting the BPM cycle by a software system, Section 4.2.7). Terms and knowledge are created in a common effort, providing a continuous fusion of the contributions and a posterior quality control by the community of participants. Finally, weak ties (Section 3.1) support also positively semantic integration through fostering collaboration Section 4.2.3, and reconciliating different terminologies, Section 4.2.5.

4.1.3 Responsiveness

To increase the responsiveness, fostering the motivation to participate (Section 4.2.1), empowering human agents at the micro level (Section 4.2.4), extending notations to represent both social and business processes (Section 4.2.6) and supporting the BPM cycle in a BPM system 4.2.7 are possible manners/approaches. Weak ties are a feature which increases responsiveness by bypassing hierarchy**.** Social production supports responsiveness by ad-hoc planning of production and the visibility of intermediate results. The reaction to an event does not follow pre-defined procedures but is controlled in an ad-hoc manner. The logging of all steps and the collective evaluation by all stakeholders also assures the quality of the solution provided. Furthermore, the visibility of intermediate results ("perpetual beta") allows the overlapping of tasks and thus the reduction of reaction times. Mutual service provisioning supports the

responsiveness necessary for agile BPM by allowing the interweaving of tasks and the avoidance of a strict ordering of tasks.

4.2 Key Approaches for Enabling Agile BPM with Social Software

4.2.1 Fostering Motivation to Participate

4.2.1.1 Motivation

Organisational integration (Section 4.1.1) may be less effective due to a lack of motivation to participate. In fact, a BPM lifecycle based on social software requires the collaboration between different stakeholders through the different phases of the BPM lifecycle. However, stakeholders may lack a commonly perceived identity and this may hamper participation (for a discussion about the influence of a common identity on motivation see e.g., [23]).

It is possible to observe this fact by considering what usually happens with one of the most frequently used technologies to specify the implementation and execution of business process models, i.e., Workflow Management Systems (WfMSs) [31]. Workflow Management Systems provide users with a business process representation, that is, a set of formal rules with which they have to comply. Rules embedded in the business processes are defined by controllers (e.g., managers of the organisation) to ensure that individuals act consistently with the organisational goals and objectives. In this view WfMSs are tools for exerting a formal mode of control [32]. Consequently, conflicts among controllers and controlees may arise especially when the rules proposed by controllers change controlees' consolidated work practices.

4.2.1.2 Approach

In a precedent study on Wikipedia [32], one of the most widely used pieces of Social Software, strategies Wikipedia users adopted to manage conflicts were identified. Those strategies basically consider the construction and the management of a common identity among participants. In [32] it was argued that the perception of a common identity would also help users in forming a positive attitude towards the business process representation mediated by the workflow system. In fact, in this way, both controllers and controlees are perceived as part of the same group, sharing objectives, beliefs, intentions, fate and also norms and rules expressed by the process representation. In order to facilitate both controllers and controlees in the creation and maintenance of a common group identity, controlees should be allowed to express their opinion concerning the way the business process system regulates their work activities. This would help them to perceive that their contribution is fundamental for the achievement of the organisation goals, increasing their motivation and their commitment to it.

4.2.1.3 Related work

The literature is rich of contributions about the integration of workflow technology with Social Software. In this respect, literature presents some proposals to combine functionalities of Social Software, such as flexibility and integrated revision management of Wikis, with Workflow Management Systems. In , for instance, the principles underlying workflow management (like the modeling of processes and their execution) have been implemented in terms of a Wiki system. In particular, most of the proposals focus specifically on the collaborative design of workflow

processes (e.g., [33], [34], and [35]). The point here is different, since the focus of the approach is not on the issue of designing a process in a collaborative way, rather it is on defining a set of mechanisms fostering the motivation of people to design processes in a collaborative way. So in this approach we consider a socio-psychologically perspective, in particular the Social identity approach, and we apply this to the design of tools supporting organizational integration.

4.2.1.4 Contribution and integration into overall solution

Social Software used for overcoming organisational barriers (Section 4.1.1) has to play a further role (i) to make organisational integration more effective; (ii) to facilitate the ability to create and (iii) to maintain a common identity among all the stakeholders involved in the different phases of BPM lifecycle. This would limit the negative impacts of conflicts arising from different stakeholders' points of view. In addition, motivation to participate should also be leveraged and hence organisational integration should become more effective. Furthermore, in the long run, this support should also facilitate the definition of more effective change management strategies and hence enhance the responsiveness of the BPM lifecycle (Section 4.1.3). The mutual service provision feature of social software (Section 3.4) has also a positive effect on fostering motivation to participate.

4.2.2 Sharing Knowledge

4.2.2.1 Motivation

As stated earlier, agile BPM relies on the involvement of stakeholders in the improvement and adaptation of business processes. Therefore, another important point for overcoming organisational barriers is the dissemination of knowledge of business processes within an organisation and among users. The key for this dissemination is to make business process models available to relevant stakeholders. Social software can support this aim by using tagging mechanisms to describe and distribute business process models.

Nowadays, process models are standard tools for capturing business processes in organisations. Thus, they have great potential for the sharing of knowledge in organisations. By graphically representing those processes, they make this knowledge available to different stakeholders. Unfortunately, a series of interviews with practitioners using models in their companies [36] shows that this usage potential is widely neglected in organisations:

- Model usage in organisations is typically restricted to a core of a few people such as analysts, developers and managerial staff, who use BPM systems to manage business processes and their corresponding models. Other people such as business process participants or new employees who need to know about organisational processes or want to give feedback on business processes are more or less unaware of models as sources for information on business processes. Obviously, this is a situation that is not sufficient to overcome organisational barriers.

- In repositories accessible for all stakeholders in an organisation, such as knowledge management systems, models are usually neglected. This is because such systems cannot interpret the content of models and therefore manage models only by their file name. Therefore, it is often not possible to find models in such systems. As a consequence, the problem of scant knowledge on business processes and the lack of feedback get worse.

- The majority of staff in organisations does not accept the use of models. They often perceive models as technical artefacts used by modellers and analysts but not as valuable means for knowledge sharing. On the other hand, models are closely bound to the modellers, who are in control of changes to them, thus making other staff bystanders in process documentation and change. Getting feedback for business process models is costly and has to be done by modellers personally asking people for their feedback.

Analysing this situation, a vicious circle of the lack of availability of business process models, of scarce knowledge on business processes and of the lack of feedback on them can be identified. If models can be disseminated throughout a whole organisation, people will be able to perceive them as valuable content, acceptance is likely to rise and the circle can be broken. For this to happen, models should be integrated into knowledge management systems, which are used by the majority of staff in organisations (as opposed to BPM systems). This can make models available to all people in an organisation.

4.2.2.2 Approach: Integration of process models into organizational knowledge management

Social tagging mechanisms provide an opportunity to accomplish such integration as they enable the description of a model's content, thus making the content available for search mechanisms and for people browsing knowledge management systems in search of information. The potential and advantages of social tagging in explicating knowledge captured in process models are manifold (cf. [37]):

- First, they enable users to express easily their understanding of a business process model by assigning tags to it. This also enables them to find a model when they need it sometime later on.
- Second, if multiple people tag a model, perspectives on the model can be identified by tag clustering and thus the model can serve as a boundary object [38] fostering conversations between groups. Implementing egalitarianism (cf. Section 3.3), tags can be provided by all members of an organisation and thus provide a content description transcending hierarchies and boundaries e.g. between divisions.
- Third, tags contextualise process models and therefore enable users to see both the topics covered in models and, by searching for content marked with similar tags, to find additional information on a certain topic. Bearing in mind that all people are allowed to provide tags as described above, this context provided by tags is created socially (cf. Section 3.2) and thus represents perspectives of many different stakeholders.
- Fourth, by analysing tags associated with a model, a knowledge management application can find people who are knowledgeable about the corresponding process and might thus be able to give information on this business process (cf. [39] for a similar approach).

Taken together, integrating business process models into knowledge management systems by using tagging mechanisms brings business processes to people in organisations. This helps to establish models as artefacts of daily work and knowledge acquisition. Additionally, it fosters business process improvement. If process models are accepted and available in organisations feedback on business processes is likely to increase. Therefore, organisational goals such as continuous process improvement will benefit from the dissemination of business process models.

This helps to overcome organisational barriers in Agile BPM. To show the possible application of tagging mechanisms for the potentials mentioned above, a prototype using a tagging mechanism for process models has been created and evaluated (cf. [36]). The evaluation was conducted with 10 people. Of these people were 8 researchers and 2 students using process maps for process analysis, consulting, design and software implementation for/with companies they work with. The workshops were conducted, but had a lot of free space for the participants to do what they wanted to do. Initial results of the evaluation show that the prototype enables communication on business processes, which in turn enables users to get involved in business process adaptation. Additionally, the evaluation shows that it contextualises business processes with corresponding knowledge sources and thus makes them more understandable for stakeholders. Furthermore, people felt that it recommends relevant models to certain stakeholders and thus implicitly calls on them to get involved.

4.2.2.3 Related Work

As mentioned above, the perspective of actively using process models as sources of knowledge and artefacts to document knowledge for the whole organization has not been looked into deeply. There are systems enabling the dissemination of process models such as [40]. Additionally, there are approaches in knowledge management, which use process models for process guidance and orientation in complex content areas [41], [42]. Neither of these approaches, however, presents solutions to sharing and using process models as knowledge artefacts.

Recent contributions to the community show how sharing the knowledge documented in process models and documenting knowledge in process models can be fostered. Among these, Koschmider et al. show how process models can be shared and exchanged based on social network proximity of modellers [35]. Additionally, there are approaches enabling users to publish their models on the web, contextualize them and make them accessible to their community [43]. Other interesting approaches can be found in using social tagging mechanisms for relating models dynamically [44] or managing them in a model repository [45].

The abovementioned approaches provide valuable solution information on how to use process models for knowledge exchange. However, there is no comprehensive solution for the needs described above. In the remainder of this subsection, we will therefore describe an approach and its prototypical implementation of such a solution.

4.2.2.4 Contribution and integration into overall solution

Despite the results of the evaluation, the prototype should not be thought of as designed as a standalone version or as a replacement for BPM systems. The tagging of models is intended to bring models *already existing and stored in BPM systems* to people in a more accessible manner. This can be achieved if a knowledge management system and a BPM system *share* models as their content. Using such an approach, both systems and their strengths can be used and the knowledge management system *supports* awareness of processes as well as the involvement of stakeholders in process design and adaptation.

Taken together, sharing knowledge provides also a way for fostering motivation to participate (cf. Section 4.2.1) and fostering collaboration in process design (cf. Section 4.2.3). By providing content description transcending hierarchies and thus implementing egalitarianism (Section 3.3),

social tagging can also distribute process models throughout all levels of an organisation. Therefore, it supports organisational integration. Moreover, tagging results in rich descriptions of business process models which are produced socially by many stakeholders and therefore may support overcoming the semantic barriers existing in BPM (cf. Section 4.1.2). Besides these benefits, tagging may also support semantic integration (Section 4.1.2) by grouping different terms (tags) provided by users and representing different terminology used by them, making these terms explicit on processes and thus both make visible and span boundaries between different groups of stakeholders.

4.2.3 Fostering Collaboration

4.2.3.1 Motivation

The idea of fostering the collaboration among BPM stakeholders is not new. BPM collaborative modeling environments, e.g., [46] and [47], are intended to facilitate the participation of end users in business processes modeling activities. However, end users are required to work at the expert's level of abstraction using BPM expert languages, mainly BPMN. On the other hand, end users often have a local perspective of a given business process: they know which business activities to execute and their organisational responsibilities, but they are not aware of the overall goals and structure. Conversely, experts' modeling activities (part of BPM lifecycle and the underlying engineering processes) are driven by systemic soft goals, like optimisation, which are transparent to end users. As a result, the conceptual gap between the perspective of the expert and the end users hinders the capture of tacit knowledge of such users when they model business processes collaboratively.

4.2.3.2 Collaboration among unequals

In this section, a strategy is proposing to enable an effective participation of end users in the modeling of business processes at their own level of abstraction. This strategy stems from the assumption that end users seldom have an understanding of the complete business process. They have a local perspective, which provides them with a consistent view of their activities and does not require them to understand the whole. This consistent perspective is characterised by:

- End users know their responsibilities and the organisational environment where they work.
- When exceptional situations occur, end users are able to react consistently in the context of this particular case but seldom can picture a global consistent solution.
- When deviating from the specified business process, end users know which activities they have to perform in order to accomplish the business instance goals, but these activities may not be predefined and they may even produce information that will be necessary in a further stage of the business process instance execution.

Some business processes modeling and execution tools [48], [49] allow business process instances to deviate from the business process model to capture the tacit knowledge of the end user. However, these tools require the end user to deviate in the context of the process expert language, since the resulting instance has to be consistent. Hence, either the end users have a global perspective of the business process and master the business process expert language, or the

execution tool provides them with a limited set of possible changes that preserve the overall process instance consistency. In the latter the deviation is restricted to a limited set of options. Both situations, requiring end users to master process expert languages and limiting the set of possible changes, hinder the capture of end users tacit knowledge. In [50] a business process development environment is proposed that integrates the modeling and execution of business processes. In particular, it allows end users to deviate from the business process specification, if a non-specified situation occurs during the execution of a business process instance, by using generic (typeless) entities. For instance, they can execute a generic activity instance that consumes and produces data. In addition, it fosters the use of social software features, at the instance level, to capture the rationale behind deviations and enrich the semantics of generic entities. Both the deviation log and the informal information provided by end users through social software features embody tacit knowledge made explicit during actual business operation. Using the captured tacit knowledge, experts can decide on the evolution of the business process specification. Hence, end users' tacit knowledge is captured at the right level of abstraction and can be integrated in the business process specification by experts according to their global perspective and driven by business process systemic goals.

In order to foster stakeholders' collaboration, it is necessary to build tools which have the following properties:

- **Incompleteness**. Business processes do not need to be completely understood. Trying to completely understand a business process is time consuming, reduces the number of development cycles and increases the chances that, once implemented, the business process no longer conforms to the organisational needs.

- **Empower end users**. Business processes should promote collaboration, creativity and intelligence instead of restricting them. The system should allow end users to create and execute generic non-specified activities and combine their execution with specified activities. In addition, they can add informal and subjective information explaining the rationale behind the deviation. The resulting business process does not need to be consistent from a global point of view. A local deviation may trigger other local deviations.

- **Definition on a case-by-case basis.** It should be possible to promote unexpected behaviour, described on a case-by-case basis, to become part of the specification of the business process. This approach promotes the bottom-up definition of business processes. This bottom-up process is fostered by the informal information, produced by social software features, that semantically enriches the non-specified generic entities.

As an example, consider an online bookstore and the Selling process which has three specified activities: AddBookToOrder followed by either PayByCheck or PayByCreditCard. However, as a client goes directly to the physical store and wishes to pay in cash, there is no activity that will cover such situation. The employee may then create a non-specified activity and associate it with the current instance of the Selling process. Afterwards, the employee needs to assign this activity instance to a supervisor because they do not have enough authority to receive the money. Therefore, the employee, instead of executing the activity instance, addresses it to the supervisor. The supervisor is then able to execute the business activity instance created by the employee and

finishes the Selling process. The specified Selling process instance is terminated having two different types of activity instances: an instance of the specified AddBookToOrder activity, and an instance of a non-specified activity to cover an exceptional case of paying in cash. As can be seen, end users are empowered to perform business processes according to their tacit knowledge and responsibility can be delegated based on the roles played by the organisation members. It is up to the employee to know that in this exceptional case only the supervisor can receive the money. Both employee and supervisor can tag the non-specified activity instance with keywords such as Pay, Books, Money and Cash. In this manner, future end users who see a similar exceptional case could find similar occurrences easily while searching for those tags and make use of the same activity instance structure. Moreover, if the exceptional behaviour occurs frequently in future executions of the business process, then an expert can analyse the associated tags and comments and decide to specify the PayInCash activity. This will generate a model evolution obviating the necessity for the employee to search for similar cases: the exceptional case becomes part of the business process model.

4.2.3.3 Related Work

The work in [51] describes an integrated lifecycle that allows changes of individual process instances. Additionally, individual process instances can be enriched with contextual information about the reasons for deviations. Together, the contextual information and the deviated instances can be used in the bottom-up evolution of the process type. Their approach is similar in some aspects to agile BPM, yet there are two differences: (1) the approach presented in [50] focus on consistent global deviations, it is based on ADEPT2, hindering the capture of end users tacit knowledge, and (2) since all entities have a type, social software features are used to justify variations in the business process type while we defend a bottom-up definition of business process types through the use of informal and semi-informal representations.

4.2.3.4 Contribution and integration into overall solution

Even though constrained by their organisational context, the business process instance provides the common identity (Section 4.2.1) where stakeholders engage and that fosters collaboration. Additionally, business process tools should hold a description of the organizational structure (Section 4.1.1) to allow ad hoc communication mechanisms integrated in the business process modeling and execution, e.g. delegation of activities or sending a request to members of the organization to rate and comment a particular business process model.

On the other hand, the association of tags to business process instances provides a shared knowledge which helps to bridge the gap between end users informal tacit knowledge and process types. Therefore, semantic integration (Section 4.1.2) can be achieved by providing a continuous knowledge base from informal to formal knowledge.

Egalitarianism (Section 3.3) broadens the BPM lifecycle, so that it involves all stakeholders of processes. However, egalitarianism should preserve stakeholder's different languages and perspectives. Hence, the informal and subjective information resulting from the use of social software promotes collaboration as it helps to bridge the gap between stakeholders' languages and perspectives.

4.2.4 Empowering Human Agents at the Micro Level

Section 4.2.3 discussed organisational integration (Section 4.1.1) in the context of modeling environments. This section looks at a different aspect of human agent interaction and empowerment within business process, that of integration.

4.2.4.1 Motivation

The lifecycle of a specific business process requires, in the main, human agent integration at two general junctures; expert decisions and, as mentioned in Section 3, unanticipated exceptions. In the enterprise, the initial application area of Service Orientated Architecture (SOA) in relation to BPM, focused on machine to machine integration and has been recently extended to consider human agents. Such considerations come from hierarchical business structures, built upon a priori data. The dichotomy of this methodology creates a schism between the desire to prescribe specific models and the stipulation for flexibility. By leveraging social software analysis, a different approach is possible, revealing a more subtle manner of lightweight ad hoc processes, or "Flexible Micro Workflow" (FMW).

4.2.4.2 Approach

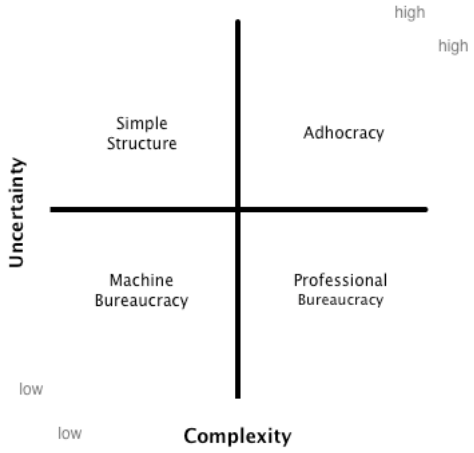

Figure 3 : Picture of high complexity and high uncertainty, based on work from [52]

Figure 3 shows a quadrant with the varying characteristics of uncertainty and complexity. In an adhocracy [53], a domain with both high complexity and uncertainty, there is inherently the lack of a priori knowledge, as outlined for the social production feature of social software (Section 3.2) , and therefore the need for a different approach to business processes, via the concept of FMW. By creating a mechanism which will support such lightweight, ad hoc, quick fire human to human interactions, a different kind of flexible workflow can be revealed. In order to support such lightweight mechanisms, an agile categorisation process is required. Prototype Theory puts forward the position of looking at base-level categories rather than classical hierarchies as "Most, if not all, categories do not have clear-cut boundaries" [54]. This position strikes clear resonance with current bottom up folksonomical strategies [55]. FMW looks to extend this with the application of social analysis, moving it into a multidimensional space [56]. Rather than finding a specific archetype of a business process or deriving an abstracted classification of a workflow

with engaged human agents as an ancillary concept, FMW looks to extend the Prototype Theory notion further. In expertise driven problem domains, building information around specific human agents enacting a given process, in the form of a layer of reputational meta data, will enable more flexible solutions when finding the most appropriate human agent for any given process. Reputation in the context of FMW refers to a body of social data which can be acquired, analysed and represented programmatically via a framework called Reputation-based Message Routing. Such weak-ties data may be mined from wikis - as mentioned in Section 4.2.1 (fostering motivation to participate), blogs, group ticking systems, cvs, mailing lists and micro messaging services such as Twitter. The details of such a service fall outside the scope of this paper but are described in further detail elsewhere [57].

4.2.4.3 Related work

The traditional SOA concept of human agent flexibility, ad hoc workflows, tends to have a fixed concept space [48]. Supporting frameworks from this area are normally based on the premise of an agent being able to either pick or create a sub workflow, or to delegate an assigned task. All such selection being made within the bounds of predetermined agent roles. Standard business practice methodology comes from modeling, process mining or hierarchical creation. With design time abstractions, time is taken by the workflow expert to ascertain the generalised, abstracted workflow model and, from that knowledge of the business process, to create the model or hierarchy. In expert driven domains, where elements of expertise are required, rather than treating solely a step or node in a pre-modelled workflow as an atomic unit, FMW suggest that there are many exploratory interactions, such as spontaneous creation facilitated by weak ties (Section 3.1), which occur but are ignored by standard approaches, as they require a priori knowledge. To build on the concepts of requiring expert decisions and unanticipated exceptions, FMW uses the core features of no reliance on a priori knowledge and of assisting such domain experts in the execution of their work providing a way to responsiveness (Section 4.1.3). As such, a FMW may be defined as an expansion of a hitherto opaque node, within an exploratory domain complex workflow, whereby lightweight non-deterministic sub-process human agent interaction occurs, such as to facilitate the successful completion of said node. This may be seen as an act of abolishing a hierarchical structure as described in section Egalitarianism.

4.2.4.4 Contribution and integration into overall solution

Supporting the execution of ad hoc business processes poses two fundamental classification questions: what is the nature of the task? And who is the most appropriate human agent to execute such a task? The paradigm proposed by FMW, in the context of human agents in an expert driven domain, effectively creates a many to many mapping. This concept moves the question from that of a predetermined hierarchy created by a small group of people, through the thought process of many possible types of a class, to the social state of an ecosystem of opinions on the nature of such a class. Just as tagging, as referenced as a way of sharing knowledge (Section 4.2.2) via folksonomies gives a greater degree of flexibility to providing Meta data over that of formal ontologies [58], the FMW paradigm suggests a similar many to many relationships. Formal ontological work endeavours to find the one best classification for a specific object by an individual, or group, of experts. Folksonomies suggest that many classifications by many people provide greater flexibility and insight into the objects and process. This capturing of

classifications as part of the lifecycle of the Flexible Micro Workflow, may be seen as one implementation of submitted requirements for the organisational integration (Section 4.1.1). Flexible Micro Workflows, rather than relying on a formal, hierarchical structuring of what is the one best workflow pattern, suggest a multi-nodal, free flowing style of interaction provided by the human agents carrying out the business process. The inverted concept behind FMW addresses the business requirements for increased flexibility in ad hoc processes and the desire to leverage social knowledge in wisdom of crowd's manner [24]. The next section will look at an extension to one of these rich sources of expert human agent generated data, wikis and how such an extension can assist in mining data which could, in turn be leveraged in the adaptive categorisation concepts mentioned above.

4.2.5 Reconciliating different Terminologies

4.2.5.1 Motivation

Even if the organizational environment fosters the stakeholders to contribute during the phases of BPM life cycle (cf. Section 4.2.1) they are partially excluded because the terms and models used to describe processes are defined without their participation or simply imposed on them. Research on cross-organizational business processes has shown that involved stakeholders have different ways to communicate and use many different modeling languages [59] . This given fact also exists in the BPM lifecycle within companies. Involved stakeholders use their own language and terminologies that they are familiar with in the BPM lifecycle.

4.2.5.2 Approach: Semantic Integration using Semantic MediaWiki

As stated earlier, agile BPM relies on a common used terminology that misunderstandings, which may prolong the BPM lifecycle, can be eliminated. In order to address the issues of different terminologies, this sub-section discusses the use of a Semantic MediaWiki (SMW) [60] that allows semantic integration (cf. Section 4.1.2). SMW extends the MediaWiki software, which runs the popular Wikipedia site. The extension combines the collaborative aspects of wikis [61] with Semantic Web technology to enable large-scale and inter-departmental collaboration on knowledge structures. Users are spontaneously connected to each other based on the feature of weak ties (Section 3.1) by creating and editing new views on problems and solutions in SMW. They combine their competencies by expressing their knowledge with natural language combined with formal annotations allowing machines to process such knowledge. For this purpose, SMW enables the user to define concept hierarchies and semantic properties related to wiki pages. SMW stores this expressed knowledge and makes it easily accessible for all users by providing an inline query language. This language enables users to query for semantic properties and concepts. In addition, users can also access older versions of processes and terminology and compare them by using the versioning functionality offered by SMW. The template functionality of MediaWiki[2] and existing extensions for SMW like Semantic Forms[3], which provides more user-friendly forms to enter semantic annotations into SMW can also be used to enhance functionality. To provide access to the stored knowledge, SMW offers RDF export functionalities, which is a standard model for data exchange on the Web.

[2] http://www.mediawiki.org/wiki/Help:Templates

[3] http://www.mediawiki.org/wiki/Extension:Semantic_Forms

SMW can also act as a repository for business processes that can be reused for specifying new business processes and collaboratively refined by the parties involved. The approach described in [33] can be used to model processes within SMW. Each process activity can be represented by an activity and the flow between activities can be expressed by using special predefined process properties. The methodology using SMW for developing processes is based on the Knowledge Maturing Process Model [62]. The model structures five phases for the maturation of knowledge building upon each other: the expression of ideas, the distribution in communities, formalization, ad-hoc learning and standardization. Regarding the maturation of business process models especially with knowledge-intensive ones, the conceptual model proposes to start with individual task lists and routines. Task patterns can be derived from recurring tasks and shared between individuals. In a next step, more stakeholders are involved, which discuss, refine, enhance and complete these procedures in a social and collaborative manner. Existing business processes can easily be linked to emerging processes and refined if necessary. From this basis, information reuse is possible. This approach of collaborative process development and sharing using SMW including import and export of process activities into/from SMW has been used and evaluated within the ACTIVE project[4].

4.2.5.3 Related Work

The Moki enterprise [63] This approach enables collaborative enterprise modeling. It focuses on the construction of a structured description of relevant aspects of an enterprise. It is another approach how SMW can be used to define a common terminology.

4.2.5.4 Contribution and integration into overall solution

The stakeholders are able to use natural language to contribute in combination with formal annotations and do not have to learn 'another' process modeling language. Thus the stakeholder describes the business process with their own language and formalise it by adding concepts and semantic properties within the text. The proposed approach lowers semantic barriers (Section 4.1.2) that exist through the use of a formalised modeling tool. Users in companies are familiar with the wiki syntax because it is easy to apply and many of them have used it before. Users are no longer only "consumers" who are forced to accept the business processes created for them (e.g. by externals) but active participants (cf. Section 3.3). In addition to this observation, all users can contribute to the definition of terms in a collaborative manner that can be realised by additional wiki pages, like a glossary, or by using the discussion functionality provided by SMW. Thus users can discuss and evolve the meaning of terms.

All users can access the SMW and thus participate in the life cycle of the business processes in which they are stakeholders. Thus, the use of SMW enables social production (Section 3.2). The knowledge created and stored in SMW can also be shared (cf. Section 4.2.2) and used by other applications and for other purposes. As a result the BPM lifecycle becomes more responsive (cf. Section 4.1.3).

As a conclusion, the social production feature of SMW can support reconciliation of different terminologies to lower semantic barriers by addressing most of the deficiencies of existing

[4] ACTIVE - Enabling the Knowledge Powered Enterprise (http://active-project.eu/) is the EU FP7 Integrating Project

approaches. Hence, misunderstandings caused by different terminologies can be eliminated. Another advantage of using SMW is that it realises egalitarianism by allowing every stakeholder to contribute to BPM lifecycle and consume resulting information.

4.2.6 Extending notations to represent both social and business processes

This section presents a short survey of notations for business processes and proposes the introduction of a new kind of building block in order to deal with cooperative actions.

4.2.6.1 Motivation

A business process is a series of steps designed to produce a product or service [64]. Let us underline the fact that this definition corresponds to a well-defined business process and not at all to a knowledge-intensive one. Nevertheless, it will be our starting point in this sub-section taking into account that numerous research works took the party of enhancing the activity-driven models by making the control flow less rigid than changing drastically the representation perspective [10][65]. Most business processes are cross-functional in that they involve members of different departments. Efficiency is the major goal, which is achieved by means of a rigid control flow. Business processes are usually inspired to a centralized perspective: they present the viewpoint of a central authority having the power of distributing the work to the members of the organization. A business process is then a global model of work, i.e. a centralized description of the steps of work (or tasks) to be carried out by the various members subdivided in roles.

As an example, Figure 4.a shows a portion of a business process meant to reply to requests for quote (Rfq) coming from customers. This process, called RfqHandlerBP, is run by a selling organization. The notation used draws on BPMN 2.0. The roles involved are shown within parentheses. The first task, checkRfq, is carried out by the sales manager and consists in checking an incoming request for quote (rfq). If the check is successful, task prepareQuote is carried out by an account manager. The Rfq data object shows the data flow between the two tasks. The exclusive gateway “ok?” represents a decision point: on the basis of the outcome of task checkRfq, the process engine will continue with task prepareQuote or it will select the other branch.

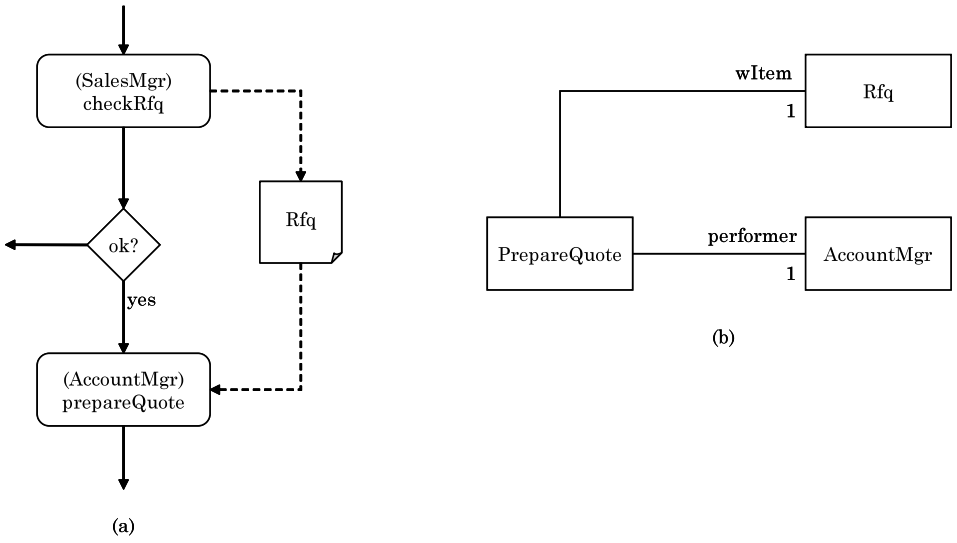

Figure 4: Portion of RfqHandlerBP (a), the assignment of task PrepareQuote (b)

In the centralised perspective (rigid control flow), participants are meant to interact with the process engine (i.e. the software component that interprets business process descriptions) and not with each other. They are presented with to-do lists showing the names of the tasks that have been assigned to them by the process engine; by clicking on the items of their to-do lists, they can perform the corresponding tasks. When a task has been completed, the next one to be carried out is indicated either directly or indirectly (through control flow elements) in the process description.

In Figure 4.b the UML class model for a task assignment is shown: the process engine assigns an instance of PrepareQuote to a particular account manager and also indicates the rfq to be worked on. Account managers and requests for quote are represented by entity classes; performer and wItem (work item) are role attributes. The performer of a PrepareQuote instance is a particular account manager and the work item is a particular rfq. To-do lists show the task assignments for the current user.

4.2.6.2 Approach: introducing cooperative activities into business processes

While it is generally accepted that rigid (well-defined) business processes have made a significant contribution to repetitive, standardised work (i.e. routines), they do not seem to be adequate to situations requiring knowledge intensive work [66].

Knowledge intensive work calls for some degree of creativity and adaptation to specific circumstances [67]. There are several directions that can be taken to make business processes less rigid. One is to make the control flow more flexible. This flexibility can be achieved if the control flow of the business process is not completely predetermined and therefore the sequence of activities to be carried out is decided at run-time on the basis of the case [68] to be handled. The case is what the process is meant to take care of; for example, it may be a patient in a hospital or the application for a job. Instead of precisely defining what is to be done, the business process includes a number of constraints so as to prescribe what cannot be done. The result is that, given a certain business process, its actual instances are likely to be variants built at run-time [69]. In

the Declare [70] framework, the run-time support shows the activities currently enabled and presents suggestions (based on historic information) to the user.

Another direction of improvement is the introduction of cooperative activities into business processes. This entails another kind of flexibility, i.e. the ability of taking advantage of the contributions of several participants without the need for an explicit orchestration. Although a process like RfqHandlerBP involves a community of participants, it cannot be really considered as a case of cooperative work in that it is made up of individual activities. Such activities are organized so as to achieve a common business goal, but the participants operate individually in different steps of the work [4].

A second version of RfqHandlerBP, called RfqHandlerBP2, is shown in Figure 5

The account manager does not directly prepare the quote; he/she, instead, first organises a wiki document to collect the suggestions of a number of advisors before proceeding with the finalisation of the quote. Then the cooperative action prepareQuote takes place.

In Figure 5.b the UML class model for the assignment of cooperative task PrepareQuote is shown. The difference from the model in Figure 5.b is the involvement of several performers (represented by instances of class Participant). These participants are assumed to have been associated with the Wiki document by the account manager in task defineWikiDoc.

Traditionally, the cooperative development of shared artefacts, such as documents and software, is considered to be part of the groupware domain because there is no explicit control flow to handle. Integrating cooperative actions in business processes promotes agility in that a number of participants can be involved quickly in a collective action without the need to define explicitly specific tasks. In a cooperative action, “group members perform sequence of actions in a shared space of actions, such that the actions of one group member can affect the space of actions of the others” [71].

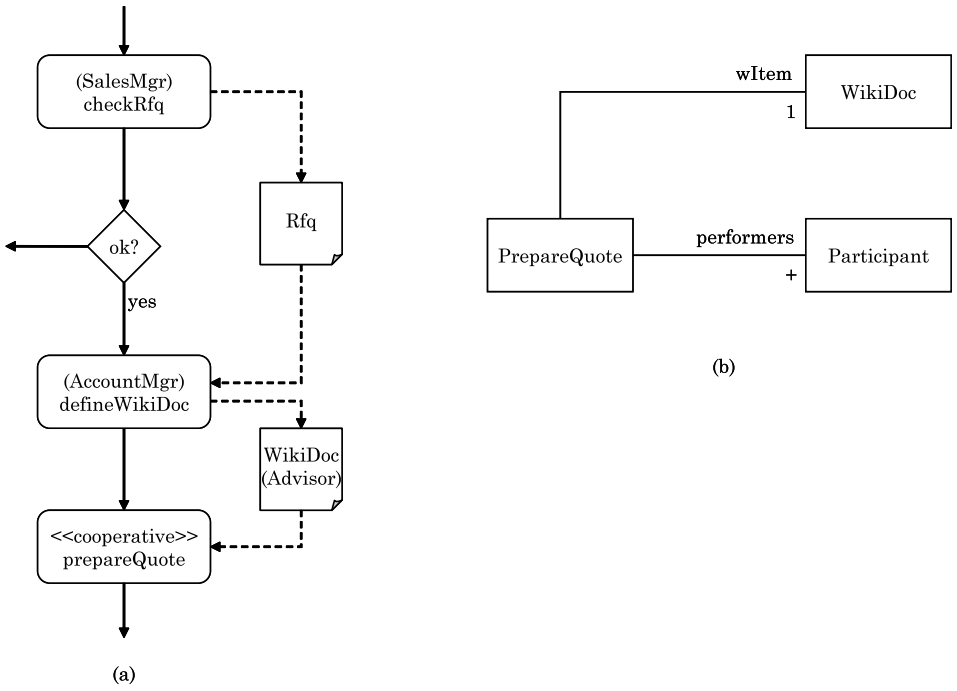

Figure 5: Portion of RfqHandlerBP2 (a), the assignment of cooperative task PrepareQuote (b)

In order to integrate business processes and cooperative actions we propose to add a new kind of building block to the standard BP elements; such process items are tagged with the <<cooperative>> stereotype as shown in 5.a. The type of cooperative entity is shown in the data object associated with the cooperative action. Term WikiDoc in Figure 5.a indicates that the cooperative object is a wiki document; the roles involved in the cooperative action, e.g. Advisor, are indicated between parentheses.

4.2.6.3 Related work

There is a growing interest in wiki systems, for their potential in terms of availability; customisability and agility in use (cf. Section 4.2.1). A wiki document is a social entity in that it enables a number of users with different roles to work out a shared document by taking advantage of the internal mechanisms provided by wiki systems e.g. access control and version management. The idea of introducing cooperative building blocks is not new: for example, macro-operations are introduced in [4]. What is to be further explored is the integration between the process engine and such cooperative actions.

When involved in a cooperative action, the participants must find a suitable assignment in their to-do list so that they can get direct access to the shared document or entity. Several cooperative tools, in addition to the cooperative production of documents, have been illustrated in the literature; some examples are COMA [72] for the collaborative development of process models and DOTS [73] for collaborative deliberations. What is needed is the availability of such

functionalities in terms of software components to be easily integrated in a process-aware information system (PAIS) [74].

A more specific example concerned with the development of a requirements definition document based on a hierarchical wiki system is presented in [75].

This section has addressed the "semantic integration" requirement of agile BPM by proposing a notation which integrates cooperative actions into business processes. Cooperative actions are a case of social production, thus they benefit from the "social production" feature of social software.

4.2.7 Supporting an agile BPM lifecycle in a BPM system

The traditional BPM lifecycle contains fixed flows between selected stakeholders, implementing an a priori control of quality. Social software, on the other hand, enables an a posteriori control of quality: All stakeholders may easily contribute but the community of stakeholders evaluates their contributions. Applying the ideas of social software on the BPM lifecycle, it follows that different users are included more actively in the BPM life cycle. Every contribution to the BPM life cycle can be seen as a service provided by an end user or an involved stakeholder.

4.2.7.1 Motivation

Supporting an agile BPM lifecycle requires overcoming the barriers to organization integration, semantic integration, and responsiveness described in section 2. An enabler for this is the more open and collaborative environment in the current trend of providing BPM “as a service”, wherein a workflow environment can be accessed by a large online community using simply a web browser. Examples include commercial systems such as Serena.com and RunMyProcess.com as well as research efforts such as the one described here. In this section, we present an end-to-end BPM as a service system that illustrates in practice how features of social software (weak ties, social production, egalitarianism (Section 3)) can be leveraged and some of the barriers to organizational integration and responsiveness can be overcome in paving the way to an agile BPM lifecycle.

4.2.7.2 Approach

The system, detailed in [76], [77], [78] is a hosted lightweight workflow system (WfMS) that enables collaboration between workflow designers who create processes, end users who use them and software developers who create programs which extend existing workflow systems and languages. It consists of a workflow editor for creating workflows, a runtime for executing them, a catalogue of extension activities and a repository of extension activity implementations. The system enables collaboration and sharing with a focus on extension activities, workflow models (also referred to as business process models in previous sections), and workflow instances. An extension activity is a special kind of workflow activity that provides functionality that is not a built-in part of the workflow language or system in use. It is usually created by a software developer as it requires programming language skills.

The system enables social production of extension activities by providing developers the ability to create, immediately share, and dynamically deploy extension activities into the overall design and runtime environment. By using the simple REST-friendly Bite workflow language [79], [80] the steps of creating extensions and providing them to the runtime in this system are greatly simplified from other workflow systems (as shown in [1]). A developer creates a workflow extension activity by extending a known interface and then deploys it to the running system by uploading the resulting implementation artefact along with some meta data (author, tags, and functional description). They may perform the upload by either filling out a simple web form or using a plug-in to their development environment of choice. The system currently provides one such plug-in for the Eclipse development environment. Once the extension has been provided to the hosted environment, it immediately appears as an activity which can be selected by any workflow designer.

A workflow designer uses the system to create workflows by selecting and connecting a set of business activities from an extensible palette. The palette contains the built-in activities of the Bite workflow language, along with available extension activities. Each extension activity is also added to an extension activity catalogue, based on [81] , so that designers and developers can rate it, tag it, and comment on it. Once the activity is in the catalogue, its implementation is provided to the repository, enabling other designers to download it and improve it, extend it, or use it to create new related extensions which can be added to the system.

The way in which developers collaborate around extension activities has now been described. Designers interact with the catalogue to search for extension activities and can also comment, rate, and tag the catalogue's extensions. Thus, collaboration is enabled between the designers and developers. It can be seen that designers would also be able to request new extension activities from a developer marketplace. The use of this catalogue and its visibility across the designer and developer communities regardless of hierarchical and team structures thereby supports egalitarianism and creates weak ties as people come together around extensions of common interest.

Social production and collaboration are also supported for the workflow models themselves through the system's capability for designers to share and edit their workflow models with others. A workflow model can be shared at any point in its lifecycle. A designer shares a model by specifying which users or groups of users can read and/or write the model. Upon logging into the system, each workflow designer sees a dashboard with the models to which they have read access. They are then able to view and update it, if they have write permissions. The agility of the BPM lifecycle is thereby increased because process modeling by making it iterative and collaborative.

Sharing is also enabled at the workflow instance level: since the workflows are Bite flows, all their entry points are accessible via a URL. In Bite [79], a deployed workflow model becomes accessible at a base URL, which acts as an instance factory. A newly created instance gets its own instance URL, created by adding a path to the base URL. Subsequent flow instance entry points are relative to this instance URL. Thus, sharing base or instance URLs enables designers and end users instantiate a new instance and access different entry points on the same instance. In other words, the designer or user can simply provide the URL of the first entry point to share a

workflow or any entry point in a particular instance by providing the URL. The system enables design time access control on each of these entry points. Additionally, the integration with lightweight collaborative task tools such as Lotus Activities allows first class support for workflows that coordinate interactive, unstructured work done by teams of people [80]. End users are now quickly and easily able to interact with workflow instances.

Enabling this functionality in such a system is non-trivial. An in-depth explanation of the technical enablement for how these social software capabilities have been applied to BPM in this system is detailed in the [83].

4.2.7.3 Related Work

Sharing and reusability exists for several artefacts on the Web: For unstructured, Web-based artefacts is enabled via existing systems that use tagging, ranking and commenting such as for photos(Flickr), videos(YouTube) and goods(Amazon.com). Developers share applications by sharing source code in sites such as SourceForge, indexing application code (Koders) or indexing web service APIs (ProgrammableWeb) with blogs and forums surrounding the index. Social aspects of development are being added to code sharing sites such as GitHub. Social software for Web services is explored in the Wiki-based system [84] and using Service Communities [85]. Our work extends some of these concepts to the BPM domain, providing social enablement for workflows and extension activities.

Several commercial BPM-as-a-service systems exist, such as Serena Business Mashups, RunMyProcess, Lombardi Blueprint and IBM's BPM BlueWorks (merging with Lombardi Blueprint due to recent acquisition), but have a different focus and do not enable collaborative extensibility as we do here. BPM BlueWorks shares some of the infrastructure with our system.

4.2.7.4 Contribution and integration into overall solution

The BPM as a service system presented in this section extends traditional BPM by moving it into the cloud and leverages the community in order to provide an agile BPM solution. It does so by leveraging two main features of social software, organisational integration (Section 4.1.1) and increased responsiveness (Section 4.1.3), thereby collapsing the phases of the BPM development lifecycle. It overcomes some of the organisational and semantic barriers to agile BPM by enabling sharing and collaboration of workflow models, instances, and extension activities. This is done at and across the different lifecycle phases as well as between the different stakeholders.

5 Related Work

In industry, the combination of social software and business process management has already been identified as an important topic. Examples are the integration of Google Wave and BPMN based business process modeling in the Gravity project of SAP [86]. It is intended for the real-time collaboration on BPMN based business process models using Google Wave. In [35] and [63] the use of social software for modeling business processes is discussed.

There are a number of standard BPM lifecycle descriptions such as the PAIS life cycle [87], consisting of process design, implementation, enactment and diagnosis. In [88] a BPM lifecycle also with four phases is introduced: design, configuration, enactment and diagnosis phase. Those

approaches are used as basis for extensions. In [89], the use of semantic extensions of the BPM lifecycle, especially in the modeling and configuration phase, is discussed e.g. semantic process validation technologies and their application to improve model quality. Semantic-based discovery mechanisms are used to find appropriate IT infrastructure for business process implementation.

There are a number of approaches to extend the modeling of business processes in order to better capture the requirements. In [90][91] the modeling of business goals is proposed to improve the adaptability of business processes to changing business priorities or customer preferences. However, although this extends the formally represented part of reality, it does not overcome the aforementioned organisational and semantic barriers. It may be examined whether all stakeholders are able to enter their input easily and appropriately.

There are some interesting approaches which go beyond classic BPM. In [92] planning is proposed as alternative to BPM. However plans can only be applied to one given world state, but not to a changing one as in a business process. Moreover, planning relies on the assumption that the result of all actions is known in advance. A completely different approach to capture the requirements for business process is process mining [93][94]. Here, process models are not gained from interaction with stakeholders but from analysing the patterns of already executed business process instances.

Knowledge intensive processes are typically realised through weakly structured workflows [95] . There exist manifold modeling methods and tools in the research area of knowledge intensive processes for example GPO-WM [96], KMDL [97] PROMOTE [98] and DECOR [99], extending BPM methods and tools described above. Additional business activities such as information retrieval activities and perspectives such as describing knowledge required to enact an activity or knowledge flows between activities are introduced by these approaches. Fine-grained descriptions of knowledge-creation, knowledge-retrieval or knowledge-transfer activities and knowledge-oriented process analysis for designing organisational knowledge infrastructures [100] are also used.

6 Conclusion

Together with the increasing use of business process management although some shortcomings have become obvious, e.g. the model-reality divide [1] [13] appears when business process management cannot keep pace with the change requests of the process' stakeholders. Further issues are the accelerated pace of changes and the spreading of context information and the demand for quickly created process solutions. A BPM capable of reacting quickly to external and internal events is named agile BPM.

Such an agile BPM has to integrate all stakeholders on an organisational level. Furthermore, Agile BPM must be capable of homogenising the heterogeneous semantics of the requirements defined by different stakeholders. But also within the business process management lifecycle, the flow of information and the sequence of procedures have to be integrated to achieve an agile BPM.

Agile BPM does not simply require minor changes to the BPM lifecycle but a paradigmatic change. Therefore, in order to realise an agile BPM, the basic paradigm of the business process management lifecycle has to be changed. Such a change can be realised by applying social software for the support of the BPM lifecycle.

Social software offers four features, which are very valuable: weak ties, egalitarianism, social production, and mutual service provisioning. Using these features, it is possible to provide organisational and semantic integration and responsiveness as required by agile BPM. Based on this analysis, concrete approaches for realising the theoretic advantages of social software have been presented.

Organisational integration is feasible by taking into account the difficulties of involving stakeholders. Therefore, using a social identity approach, people are motivated to participate if they feel part of the same group and have developed a common identity. For organisational integration, stakeholders have to be aware of business processes (as well as BPM processes to some extent), the corresponding knowledge and the possibility of participating in the BPM lifecycle. Therefore, business processes knowledge and models depicting those processes should be available in organisational knowledge management systems. Business processes are published among staff and input as well as feedback on processes can be given spontaneously and contextually.

To foster collaboration among stakeholders, it is important to allow them to communicate using their own perspectives and languages. The use of social software features at the instance level can be used to foster collaboration among stakeholders, while preserving their perspectives and languages of the business process. Agility has been shown to be a key concern particularly when utilising expert human agents. Human agents have been shown to generate many social interactions in new collaborative tools. This weak-tie data can be used as a basis for new classification concepts and leveraged in Flexible Micro Workflows to enable more agility for expert human agents in BPM.

Semantic integration is achieved by the fusion of the worldviews of stakeholders in the early phases of requirements elicitation. To be successful, the creation of a common language has to be done in a multitude of small steps, giving the each stakeholder to smoothly adapt a common usage of terms. Using social software, the definition of terms and processes is transformed into social production, allowing all stakeholders to contribute. To this aim we need to consider techniques to facilitate the reconciliation of multiple terminologies and the representation of both social and business processes. In order to illustrate these concepts, the introduction of cooperative activities (aimed at producing shared artefacts) into business processes has been shown. This provides the ability of collecting the contributions of several participants without the need for an explicit orchestration. Further work will be devoted to two lines of development. The first line is concerned with the definition of a suitable personal workspace in which each participant can perform his actions and can observe the results of the actions of the other participants. The second line of development is to investigate how to run social processes on social networks by taking advantage of the dynamic teams and the informal interactions they provide. Traditional BPM lifecycles are based on the assumption that it is possible to predefine the control and information flows needed during the BPM lifecycle. The intention is to ensure a certain quality

by allowing only predefined individuals to perform tasks in the lifecycle. Social software breaks with this assumption and replaces the a-priori control of quality by an a posteriori one. Many more individuals are allowed to contribute but the community of stakeholders evaluates their contributions. Therefore, end users are more included in the BPM life cycle and play an active part of the BPM life cycle. Furthermore, there is no strict separation between contributor and consumer. Instead there is an exchange of contributions and each contribution to the BPM life cycle can be seen as a service provided by an end user or an involved stakeholder. To support these concepts, a method and system are presented for supporting an agile BPM lifecycle in a BPM system[83]. The system provides a hosted BPM as a service environment, which allows sharing and collaboration around different workflow artefacts between designer, developers and end users participating in the BPM lifecycle.

The idea of personal task-knowledge management which can be evolved into reusable and shareable task patterns[101], [101][102], [103] fits well into parts of the social software approach. Ad-hoc processes for knowledge-intensive tasks are analysed in this idea. The combination of both ideas may be subject to future work.

To fully leverage the benefits of social software for an agile BPM, the introduction of social software has to be embedded into a change of corporate culture. There has to be a change from a culture of suspiciousness to a culture of trust. Before social software, enterprise organization intended to achieve quality by introducing a huge amount of a-priori quality checks. Thus in a top-down manner, rules for all participants have been defined in order to achieve a high-quality output. The success of this approach is as good as the rules defined. However, in very complex business processes it is not even possible to define such a set of rules a priori. Here, social software may help due to it's a-posteriori approach. Deficits in quality damage the reputation of the individual. Therefore, the individual will do its very best to achieve a maximum degree of quality.

This mechanism works only as long as bad quality really damages the reputation of the individual. However, the use of social software may lead to bad quality, if this mechanism is put out of order. There are a number of possible reasons. E.g. responsibility can be declined if there is a lack of transparency or if there fractions within the individuals.

Another risk created by trading control for trust is the loss of confidentiality [104]. In an environment where everybody may provide information, confidential information can be revealed already by the combination of pieces of information that are unproblematic on their own.

In [105] three financial risks of social software are identified. First, the financial return from using social software may be overestimated. Not every collaboration creates huge financial benefits. Second, opportunity costs have to be taken into account. They measure the value which could have been created when relinquishing using social software. Third, the collaboration costs may not be underestimated. Employees may demand for financial incentives compensating the extra work due to using social software.

7 References

[1] R. Schmidt and S. Nurcan, "BPM and Social Software," in *Business Process Management Workshops*, 2009, pp. 649-658.

[2] K. Schmidt and C. Simone, "Coordination mechanisms: towards conceptual foundation of CSCW systems design," *Computer Supported Cooperative Work*, vol. 5, pp. 155-200, 1996.

[3] S. Nurcan, "A method for cooperative information systems analysis and design: CISAD," in *Proceedings of the Second International Conference on the Design of Cooperative Systems (COOP'96)*, pp. 12-14, 1996.

[4] S. Nurcan, "Analysis and design of co-operative work processes: a framework," *Information and software technology*, vol. 40, no. 3, pp. 143-156, 1998.

[5] I. Bider, T. Halpin, J. Krogstie, S. Nurcan, E. Proper, and R. Schmidt, *Enterprise, Business-Process and Information Systems Modeling*. Springer, 2010.

[6] "Business Process Modeling Notation, V.1.1," OMG, 2008.

[7] "Web Services Business Process Execution Language, V.2.0," OASIS, 2007.

[8] S. Sadiq, W. Sadiq, and M. Orlowska, "Pockets of flexibility in workflow specification," *Conceptual Modeling—ER 2001*, pp. 513-526, 2001.

[9] M. Pešic, "Constraint-based workflow management systems: shifting control to users," PhD thesis, Technical University Eindhoven, 8, 2008.

[10] S. Nurcan, "A survey on the flexibility requirements related to business processes and modeling artifacts," in *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, vol. 8, pp. 7-10, 2008.

[11] T. Winograd, "A language/action perspective on the design of cooperative work," in *Proceedings of the 1986 ACM conference on Computer-supported cooperative work*, pp. 203-220, 1986.

[12] Y. Sakamoto and E. Kuwana, "Toward integrated support of synchronous and asynchronous communication in cooperative work: An empirical study of real group communication," in *Proceedings of the conference on Organizational computing systems*, pp. 90-97, 1993.

[13] C. Richardson, "Social Technologies Will Drive The Next Wave Of BPM Suites," *The Forrester Blog For Business Process & Applications Professionals*. [Online]. Available: http://blogs.forrester.com/business_process/2009/09/social-technologies-will-drive-the-next-wave-of-bpm-suites.html. [Accessed: 29-Nov-2009].

[14] F. A. Cummins, *Building the agile enterprise: with SOA, BPM and MBM*. Morgan Kaufmann Pub, 2008.

[15] H. Mintzberg and A. McHugh, "Strategy formation in an adhocracy," *Administrative Science Quarterly*, pp. 160-197, 1985.

[16] S. Nurcan, R. Schmidt, and P. Soffer, "BPMDS'08 Business Process Life-Cycle: Design, Deployment, Operation & Evaluation," in *CAiSE'08 Workshop Proceedings*, 2008.

[17] R. Ukor and A. Carpenter, "On Modelled Flexibility and Service Selection Optimisation," in *9th Workshop on Business Process Modeling, Development and Support, Montpellier, France (June 2008)*.

[18] K. Bessai, B. Claudepierre, O. Saidani, and S. Nurcan, “Context-aware Business Process Evaluation and Redesign,” *Business Process Life-Cycle: Design, Deployment, Operation & Evaluation (BPMDS’08)*, 2008.
[19] S. Erol et al., “Combining BPM and Social Software: Contradiction or Chance?,” 2009.
[20] Y. Benkler, *The Wealth of Networks : How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.
[21] D. Tapscott and A. Williams, *Wikinomics: How Mass Collaboration Changes Everything*. {Portfolio Hardcover}, 2006.
[22] S. Nurcan and R. Schmidt, “Workshop on Business Process Management and Social Software,” *http://crinfo.univ-paris1.fr/users/nurcan/BPMS2_2009/*. [Online]. Available: http://crinfo.univ-paris1.fr/users/nurcan/BPMS2_2009/. [Accessed: 25-Jan-2010].
[23] S. Alexander Haslam, *Psychology in organizations - The Social Identity Approach*. SAGE Publications, 2004.
[24] J. Surowiecki, *The Wisdom of Crowds*. Anchor, 2005.
[25] M. Granovetter, “The strength of weak ties: A network theory revisited,” *Sociological Theory*, vol. 1, no. 1, pp. 201-233, 1983.
[26] S. Nurcan, “Main concepts for cooperative work place analysis,” in *Proceedings of the Telecooperation Conference of the 15th IFIP World Computer Congress*, 1998.
[27] F. Dengler, S. Lamparter, M. Hefke, and A. Abecker, “Collaborative process development using semantic mediawiki,” in *Proceedings of the 5th Conference of Professional Knowledge Management, Solothurn, Switzerland*, 2009.
[28] S. L. Vargo and R. F. Lusch, “Service-dominant logic: continuing the evolution,” *Journal of the Academy of Marketing Science*, vol. 36, no. 1, pp. 1-10, 2008.
[29] J. W. Forrester and J. F. Collins, *Urban dynamics*. MIT Press Cambridge, MA, 1969.
[30] J. W. Forrester and J. Wright, *Industrial dynamics*. MIT press Cambridge, MA, 1961.
[31] K. R. Abbott and S. K. Sarin, “Experiences with Workflow Management: Issues for the next generation,” in *Conference on Computer Supported Cooperative Work*, pp. 113-120, 1994.
[32] L. S. Kirsch, “Portfolios of Control Modes and IS Project Management,” *INFORMATION SYSTEMS RESEARCH*, vol. 8, no. 3, pp. 215-239, 1997.
[33] F. Dengler, S. Lamparter, M. Hefke, and A. Abecker, “Collaborative Process Development using Semantic MediaWiki,” in *Proceedings of the 5th Conference of Professional Knowledge Management. Solothurn, Switzerland*, 2009.
[34] H. Qu, J. Sun, and H. T. Jamjoom, “Scoop: Automated social recommendation in enterprise process management,” in *Services Computing, 2008. SCC'08. IEEE International Conference on*, vol. 1, pp. 101-108, 2008.
[35] A. Koschmider, M. Song, and H. A. Reijers, “Social software for modeling business processes,” in *Business Process Management Workshops*, pp. 666-677, 2009.
[36] M. Prilla, “Models, Social Tagging and Knowledge Management – A fruitful Combination for Process Improvement,” in *Proceedings of 2nd Workshop on Business Process Management and Social Software in Conjunction with the Business Process Management Conference 2009*, 2009.
[37] M. Prilla, “Semantic Integration of Process Models into Knowledge Management: A Social Tagging Approach,” in *Proceedings of BIS 08*, 2008.
[38] S. L. Star, “The structure of ill-structured solutions: boundary objects and heterogeneous distributed problem solving,” *Distributed Artificial Intelligence (Vol. 2)*, pp. 37–54, 1989.

[39] S. Farrell, T. Lau, E. Wilcox, S. Nusser, and M. Muller, “Socially augmenting employee profiles with people-tagging,” in *UIST '07: Proceedings of the 20th annual ACM symposium on User interface software and technology*, pp. 91–100, 2007.
[40] J. vom Brocke and O. Thomas, “Designing Infrastructures for Reusing Conceptional Models–A General Framework and its Application for Collaborative Reference Modelling,” in *Business Information Systems: 9th International Conference on Business Information Systems (BIS 2006); May*, pp. 501–514.
[41] P. Loucopoulos and V. Kavakli, “Enterprise knowledge management and conceptual modelling,” *Conceptual Modeling*, pp. 123-143, 1999.
[42] M. Hoffmann, T. Herrmann, M. Diefenbruch, and T. Goesmann, “PRomisE2-Recording and Displaying Situated Process In-formation in Knowledge Management Applications,” *Journal of Universal Computing. Proceedings of I_KNOW02 (11-12 July). Graz University of Technology. S*, pp. 170-176, 2002.
[43] D. De Roure et al., “myExperiment: Defining the social virtual research environment,” in *eScience, 2008. eScience'08. IEEE Fourth International Conference on*, pp. 182-189, 2009.
[44] J. Fengel, M. Rebstock, and M. Nüttgens, “Modell-Tagging zur semantischen Verlinkung heterogener Modelle,” *EMISA 2008*, p. 53.
[45] J. Reich, *Supporting the execution of knowledge intensive processes by means of expert and best- practice mediation*. München: Dr. Hut, 2008.
[46] “Lombardi Blueprint,” *Lombardi Blueprint: Process improvement is everybody's business*, 2010. [Online]. Available: http://www.lombardisoftware.com/bpm-blueprint-product.php.
[47] “The Open Model Initiative,” *The Open Model Initiative*, 2006. [Online]. Available: http://openmodels.org/. [Accessed: 11-Jan-2010].
[48] P. Dadam and M. Reichert, “The ADEPT project: a decade of research and development for robust and flexible process support,” *Computer Science - Research and Development*, vol. 23, no. 2, pp. 81-97, 2009.
[49] M. Reichert et al., “Enabling Poka-Yoke Workflows with the AristaFlow BPM Suite,” 2009. [Online]. Available: zotero://attachment/4/. [Accessed: 12-Jan-2010].
[50] A. Rito-Silva, R. Meziani, R. Magalhaes, D. Martinho, A. Aguiar, and N. Flores, “AGILIPO: Embedding Social Software Features into Business Process Tools,” in *BPM 2009 Workshops*, vol. 43, pp. 219-230, 2010.
[51] B. Weber, M. Reichert, S. Rinderle-Ma, and W. Wild, “Providing integrated life cycle support in process-aware information systems,” *International Journal of Cooperative Information Systems*, vol. 18, no. 1, pp. 115-165, 2009.
[52] H. Mintzberg, *Structure in fives: designing effective organizations*. Prentice-Hall Englewood Cliffs, NJ, 1983.
[53] A. Toffler, *Future shock*. Bantam, 1990.
[54] D. Weinberger, *Everything is miscellaneous: The power of the new digital disorder*. Macmillan, 2007.
[55] A. Mathes, “Folksonomies-cooperative classification and communication through shared metadata,” *Computer Mediated Communication*, 2004.
[56] P. Gärdenfors, *Conceptual spaces: The geometry of thought*. The MIT Press, 2004.
[57] B. Jennings and A. Finkelstein, “Flexible workflows: Reputation-based message routing,” *Proceedings of BPMDS*, vol. 8, p. 97.
[58] D. R. Millen, J. Feinberg, and B. Kerr, “Dogear: Social bookmarking in the enterprise,” in

Proceedings of the SIGCHI conference on Human Factors in computing systems, pp. 111-120, 2006.

[59] M. Born et al., "ATHENA Framework for Cross-Organizational Business Processes," in *E. di Nitto; A. Sassen; P. Traverso; A. Zwegers: At Your Service: Service-Oriented Computing From an EU Perspective*, MIT Press, 2009, p. o.A.

[60] M. Krötzsch, D. Vrandecic, M. Völkel, H. Haller, and R. Studer, "Semantic wikipedia," *Web Semantics: Science, Services and Agents on the World Wide Web*, 2007.

[61] B. Leuf and W. Cunningham, *The Wiki way: quick collaboration on the web*. Addison-Wesley, 2001.

[62] A. Schmidt, K. Hinkelmann, T. Ley, S. Lindstaedt, R. Maier, and U. Riss, "Conceptual Foundations for a Service-oriented Knowledge and Learning Architecture: Supporting Content, Process and Ontology Maturing," in *Networked Knowledge - Networked Media: Integrating Knowledge Management, New Media Technologies and Semantic Systems*, S. Schaffert, K. Tochtermann, and T. Pellegrini, Eds. Springer, 2009, pp. 79-94.

[63] M. Rospocher, C. Ghidini, V. Pammer, L. Serafini, and S. Lindstaedt, "Moki: the modelling wiki," in *Proceedings of the Fourth Workshop on Semantic Wikis (SemWiki 2009) at the 6th European Semantic Web Conference*, pp. 113-127, 2009.

[64] G. A. Rummler, A. P. Brache, and I. Books24x7, "Improving performance: How to manage the white space on the organization chart," 1995.

[65] S. Nurcan and M. H. Edme, "Intention-driven modeling for flexible workflow applications," *Software Process: Improvement and Practice*, vol. 10, no. 4, pp. 363-377, 2005.

[66] U. Riss, A. Rickayzen, H. Maus, and W. van der Aalst, "Challenges for business process and task management," *Journal of Universal Knowledge Management*, no. 2, pp. 77-100, 2005.

[67] M. Alvesson, *Knowledge work and knowledge-intensive firms*. Oxford University Press, 2004.

[68] W. M. P. Van der Aalst, M. Weske, and D. Grünbauer, "Case handling: a new paradigm for business process support," *Data & Knowledge Engineering*, vol. 53, no. 2, pp. 129-162, 2005.

[69] R. Lu, S. Sadiq, and G. Governatori, "On managing business processes variants," *Data & Knowledge Engineering*, vol. 68, no. 7, pp. 642-664, 2009.

[70] W. M. P. van der Aalst, M. Pesic, and H. Schonenberg, "Declarative workflows: Balancing between flexibility and support," *Computer Science-Research and Development*, vol. 23, no. 2, pp. 99-113, 2009.

[71] S. M. Kaplan, W. J. Tolone, A. M. Carroll, D. P. Bogia, and C. Bignoli, "Supporting collaborative software development with ConversationBuilder," *ACM SIGSOFT Software Engineering Notes*, vol. 17, no. 5, p. 20, 1992.

[72] P. Rittgen, "Collaborative modeling of business processes: a comparative case study," in *Proceedings of the 2009 ACM symposium on Applied Computing*, pp. 225-230, 2009.

[73] J. Lonchamp and F. Muller, "Computer-supported deliberations for distributed teams," *Innovative Internet Computing Systems*, pp. 167-174, 2001.

[74] M. Dumas, W. van der Aalst, and A. Ter Hofstede, *Process-aware information systems: bridging people and software through process technology*. Wiley-Blackwell, 2005.

[75] G. Bruno, "Requirements elicitation as a case of social process: an approach to its description.BPMS2'09," in *BPMS2'09 Proceedings*, vol. 43, pp. 243-254, 2010.

[76] R. Khalaf, R. Subramanian, T. Mikalsen, M. Duftler, J. Diament, and I. Silva-Lepe, “Enabling Community Participation for Workflows through Extensibility and Sharing,” 2007.
[77] R. Khalaf, “A Flow-based Approach for End to End, Keynote talk Mashups 2009 Workshop at OOPSLA 2009, Orlando, Florida.”
[78] R. Khalaf, “Designing a language and system for REST-oriented service composition as a service. Tutorial, 2nd Summer School on Service and Software Architectures, Infrastructures and Engineering. Crete, Greece, 2010,” 2010.
[79] F. Curbera, M. Duftler, R. Khalaf, and D. Lovell, “Bite: Workflow composition for the web,” *Lecture Notes in Computer Science*, vol. 4749, p. 94, 2007.
[80] F. Rosenberg, F. Curbera, M. J. Duftler, and R. Khalaf, “Composing restful services and collaborative workflows: A lightweight approach,” *IEEE Internet Computing*, pp. 24-31, 2008.
[81] I. Silva-Lepe, R. Subramanian, I. Rouvellou, T. Mikalsen, J. Diament, and A. Iyengar, “Soalive service catalog: A simplified approach to describing, discovering and composing situational enterprise services,” in *Proceedings of the 6th International Conference on Service-Oriented Computing (ICSOC)*, 2008.
[82] F. Rosenberg, R. Khalaf, M. Duftler, F. Curbera, and P. Austel, “End-to-End Security for Enterprise Mashups. In: Proc. of the International Conference on Service Oriented Computing (ICSOC'09), 24.-27. November 2009, Stockholm, Sweden.,” pp. 389-403, 2009.
[83] R. Khalaf, R. Subramanian, T. Mikalsen, M. Duftler, J. Diament, and I. Silva-Lepe, “Enabling Community Participation for Workflows through Extensibility and Sharing,” in *Business Process Management Workshops*, pp. 207-218, 2010.
[84] H. Paoli, A. Schmidt, and P. Lockemann, “User-driven semantic wiki-based business service description,” *Networked Knowledge-Networked Media*, pp. 269-283, 2009.
[85] S. Tai, N. Desai, and P. Mazzoleni, “Service communities: applications and middleware,” in *Proceedings of the 6th international workshop on Software engineering and middleware*, pp. 17-22, 2006.
[86] A. Dreiling, “Gravity – Collaborative Business Process Modelling within Google Wave,” *SAP Community Network Blogs*. [Online]. Available: http://www.sdn.sap.com/irj/scn/weblogs?blog=/pub/wlg/15618. [Accessed: 29-Nov-2009].
[87] M. Dumas, W. van der Aalst, and A. Ter Hofstede, *Process-aware information systems: bridging people and software through process technology*. Wiley-Blackwell, 2005.
[88] W. M. P. van der Aalst, A. H. M. Hofstede, and M. Weske, “Business process management: A survey,” *Lecture Notes in Computer Science*, pp. 1-12, 2003.
[89] C. Pedrinaci, J. Domingue, C. Brelage, T. Van Lessen, D. Karastoyanova, and F. Leymann, “Semantic business process management: Scaling up the management of business processes,” in *ICSC’08: Proceedings of the 2008 IEEE International Conference on Semantic Computing*, pp. 546–553.
[90] A. Lapouchnian, Y. Yu, S. Liaskos, and J. Mylopoulos, “Requirements-driven design of autonomic application software,” in *Proceedings of the 2006 conference of the Center for Advanced Studies on Collaborative research*, p. 7, 2006.
[91] G. Koliadis, A. Vranesevic, M. Bhuiyan, A. Krishna, and A. Ghose, “Combining i^* and BPMN for Business Process Model Lifecycle Management,” *LECTURE NOTES IN COMPUTER SCIENCE*, vol. 4103, p. 416, 2006.

[92] H. Ferreira and D. Ferreira, *Towards an Integrated Life-Cycle for Business Process Management based on Learning and Planning*. Technical Report, INESC Porto, 2004.
[93] W. Van der Aalst, T. Weijters, and L. Maruster, "Workflow mining: Discovering process models from event logs," *IEEE Transactions on Knowledge and Data Engineering*, pp. 1128-1142, 2004.
[94] W. M. P. Van der Aalst, B. F. Van Dongen, J. Herbst, L. Maruster, G. Schimm, and A. Weijters, "Workflow mining: A survey of issues and approaches," *Data & Knowledge Engineering*, vol. 47, no. 2, pp. 237-267, 2003.
[95] S. Schwarz, A. Abecker, H. Maus, and M. Sintek, "Anforderungen an die Workflow-Unterstützung für wissensintensive Geschäftsprozesse," in *CEUR-WS Proceedings Geschäftsprozessorientiertes Wissensmanagement – Von der Strategie zum Content*, 2001.
[96] P. Heisig, *Integration von Wissensmanagement in Geschäftsprozesse*. Berlin: eureki, 2005.
[97] N. Gronau, C. Müller, and R. Korf, "KMDL - Capturing, Analysing and Improving Knowledge-Intensive Business Processes," *J Universal Computer Science*, vol. 11, no. 4, pp. 452–472, 2005.
[98] R. Woitsch and D. Karagiannis, "Process Oriented Knowledge Management: A Service Based Approach," *J Universal Computer Science*, vol. 11, no. 4, pp. 565–588, 2005.
[99] A. Abecker, "Business-Process Oriented Knowledge Management: Concepts, Methods and Tools," Institute AIFB, University of Karlsruhe, 2004.
[100] M. Strohmaier and K. Tochtermann, "B-KIDE: A Framework and A Tool for Business Process Oriented Knowledge Infrastructure Development," *J Knowledge and Process Management*, vol. 12, no. 3, pp. 171-189, Sep. 2005.
[101] L. Van Elst, F. R. Aschoff, A. Bernardi, and S. Schwarz, "Weakly-structured workflows for knowledge-intensive tasks: An experimental evaluation," in *Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003. WET ICE 2003. Proceedings*, pp. 340-345, 2003.
[102] U. Riss, A. Rickayzen, H. Maus, and W. V. D. Aalst, "Challenges for Business Process and Task Management," *j-jukm*, no. 2, pp. 77–100, 2005.
[103] E. Ong, O. Grebner, and U. Riss, "Pattern-Based Task Management: Pattern Lifecycle and Knowledge Management - Benefits and open issues," in *Professionelles Wissensmanagement (WM-2007)*, 2007.
[104] "The Enterprise Value of Social Software - John Hagel III and John Seely Brown - John Hagel III and John Seely Brown - Harvard Business Review." [Online]. Available: http://blogs.hbr.org/bigshift/2010/09/social-software.html. [Accessed: 09-Nov-2010].
[105] M. T. Hansen, "When internal collaboration is bad for your company," *Harvard Business Review*, vol. 84, no. 3, pp. 83–88, 2009.