AGILIPO: Embedding Social Software Features into Business Process Tools

António Rito Silva[1,2,3], Rachid Meziani[4], Rodrigo Magalhães[1,3],
David Martinho[1,3], Ademar Aguiar[5] and Nuno Flores[5]

[1] Center for Organizational Design and Engineering - INOV, Rua Alves Redol 9,
Lisbon, Portugal
{rito.silva,rodrigo.magalhaes,david.martinho}@inov.pt
[2] INESC-ID, Rua Alves Redol 9, Lisbon, Portugal
[3] IST/Technical Institute of Lisbon, Av. Rovisco Pais, 1049-001, Lisbon, Portugal
[4] Laboratoire Paragraphe Université de Paris 8, Saint-Denis Cedex 02, France
rmeziani@gmail.com
[5] INESC Porto, Faculdade de Engenharia da Universidade do Porto, Rua Dr.
Roberto Frias, Porto, Portugal
{ademar.aguiar,nuno.flores}@fe.up.pt

Abstract. In today's changing environments, organizational design must take into account the fact that business processes are incomplete by nature and that they should be managed in such a way that they do not restrain human intervention. In this paper we propose the embedding of social software features, such as collaboration and wiki-like features, in the modeling and execution tools of business processes. These features will foster people empowerment in the bottom-up design and execution of business processes. We conclude this paper by identifying some research issues about the implementation of the tool and its methodological impact on Business Process Management.

Key words: Business Process Modeling and Execution, Social Software, Bottom-up Business Process Management

1 Introduction

In today's dynamic market environments, the only certainty is permanent change. The way that organizations have found to cope with such changes is to keep their business models flexible. Business models are made up of business processes and these are crucial in supporting a culture of innovation. However, if business processes are left unattended and not consciously adapted to the changing environment, they become impediments to innovation [Prahalad & Krishnan, 2008]. Since the organizations' products, which are released to the market, are generated by business processes, having them flexible is important for coping with market changes in an effective manner [Weske, 2007].

Current Business Process Management (BPM) approaches still work on the AS-IS/TO-BE paradigm, inherited from the Business Process Reengineering (BPR) era, which was widely used during the nineties. BPR is a top-down, holistic, and cross-cutting approach that takes months of analysis and impact assessment to achieve. [Morgan, 2002; Cumberlidge, 2007]. The problems identified with the AS-IS/TO-BE approaches to business process management are related to the temporal gap between the modeling and implementation phases as well as the lack of involvement of the users. These problems have created a gap between business and Information Technologies (IT), where the business has always believed that IT does not understand the semantics of business processes, while IT believes that the business has no conception on what it takes for automated business processes to execute successfully.

In this paper we focus on the human-intensive aspects of business processes, where human participation is required for activities operation, even if these activities are automated. We present here a new approach to BPM, more human-centered, following the principles of agile software development [Beck *et al.* , 2001], properly supported by a collaborative environment, and we apply them to organizational design [Magalhaes & Rito-Silva, 2009]. This research aims to define an Agile Business Process Methodology and a set of associated Tools that foster the collaborative and incremental design and implementation of work processes. This is achieved by modeling the most critical business activities first, activity by activity, and undertaking modeling and implementing business processes in a continuous cycle that receives feedback from the real use of the last implemented processes.

2 The Problem

In today's business environment, characterized by non-stop and fast occurring change, it is very hard to follow an AS-IS/TO-BE approach to Business Process Management. AS-IS/TO-BE approaches assume a complete approach to the design of business processes, meaning that it requires to completely describe business processes both AS-IS and TO-BE, before any intervention can begin (either technological or managerial). This gives rise to lengthy modeling activities aiming at capturing a complete model of both the existing business processes (AS-IS) and the new business processes (TO-BE). There are several reasons why AS-IS/TO-BE approaches do not work as well as they should:

- Different people have different perspectives on processes: top managers have a high-level perspective while users have more detailed perspectives. IT consultants, on the other hand, have a systems-slanted view of the same processes. As a result, it is difficult to get all the players to agree on what the process definitions are.
- Top-down process design is driven by the organization's institutional strategies, policies and procedures and does not take into account the tacit knowledge users deploy in operating the real organization. Type-based approaches are used in top-down process design to model abstractions that represent the common structure and behavior of several process instances. Using type-based approaches to process design disregards the representation of tacit knowledge

that is mostly gathered on a case-by-case approach. These approaches intend to capture abstractions too soon and do not capture tacit knowledge.
- The organization and its rules and structures are constantly emerging and changing. This requires intervals of (very) short duration between design, implementation and automation.
- TO-BE approaches follow mechanistic models of planned change which, as a whole, restrict collaboration and reduce the empowerment of people, disengaging them from organizational responsibilities and delegating intelligence to the Information Systems (IS).
- A business process contains many exceptions that take time to model and increase dramatically the complexity of the model (i.e. the ensemble of the business processes). Furthermore, most of these exceptions only occur in a few business process instances [Russell *et al.* , 2006; Golani *et al.* , 2007].
- A technology-free process design is a naïve approach because it ignores that there is an entangling between coordination of people and the technology used by people in the execution of the business processes. The business process design, in the absence of a technological perspective, results in processes that do not fit with organizations real praxis.

3 Requirements

Within dynamic organizations, Business Process Management should follow a new agile approach characterized by short feedback cycles [Beck *et al.* , 2001]. In their proposals about Organizational Design and Engineering, Magalhaes & Rito-Silva [2009] suggest that organizational development projects (i.e. projects involving design and engineering activities) should be planned and executed through a series of small activities of short duration, such that after each intervention a new observation is carried out to identify how the organization was changed by the last intervention. The organizational routines contained in the computer-based artefacts provide the required stability for observation points to be created. Instead of strategic alignment of IS/IT, those authors propose organizational steering. Steering emphasizes continuous analyses through observation of the organization's evolution, making small adjustments between interventions, in moving the organization towards the goals defined by the strategy. The engineering activities should be of short duration followed by the artefacts integration in the organization, where design is a continuous activity and not only a starting point but an ever changing destination.

The agile business process proposal should be characterized by:

Incompleteness

- The process does not need to be completely understood. Trying to completely understand a process is time expensive, reducing the number of feedback cycles and increasing the chances that automated processes does not conform to organizational needs.

- The process does not need to be completely specified in the sense that it is allowed that activities not pre-specified occur in some of its instances. This allows the instantaneous adaptation of process instances to the emergence of new organizational needs.
- Incompletely defined processes should be executable and its execution integrates planned and unplanned activities such that incompletely specified processes will not limit the normal operation of the organization.

Empower people

- Processes should promote collaboration, creativity and intelligence instead of restricting them.
- The system should allow people to perform unplanned activities and integrate them with planned activities.

Business process design integrated with technological usage

- To avoid a situation of paralysis by analysis due to different perspectives on processes, a modeling approach based on the operation of the business should be enforced. This way, the different perspectives on process modeling will be focused on bottom-up leveraging of the actual operation of the business.
- Integrate the execution and modeling of the process, such that the process executor is also one of its modelers, thus avoiding the shortcomings of top-down modeling of processes.

Design at the instance level

- It should be possible to describe processes on a case-by-case approach instead of trying to model all the possible situations in the process specification. This will allow a reduction of the complexity of business process models and it will provide two views of the process: the type view, containing the expected behavior common to all instances, and the instance view, containing exceptions to the expected behavior present on the type view of the process.
- It should be possible to promote unplanned exceptions, described at the instance level, to become part of the planned behavior of the business process, described at the type level. This approach promotes the bottom-up definition of processes.

4 The Proposal

Considering the set of requirements identified above, we propose an agile business process approach for bottom-up modeling and implementation of incomplete business processes. AGILIPO follows the principles of agile software development [Beck *et al.* , 2001] and of organizational design and engineering [Magalhaes & Rito-Silva, 2009]. AGILIPO is supported by collaborative modeling and execution tools that embed social software-like functionalities. The distinctive feature of AGILIPO tools is the integration of modeling with execution activities blurring the differences between definition and operation of business processes. While executing a particular instance of an incomplete business process, users are empowered to execute, on a case-by-case basis, activities that are not specified.

An incomplete process definition is specified by a set of activities that describe part, but not all, of the process instances behavior. A process instance contains activity instances which can be either specified or non-specified, where non-specified activities are called generic. An activity definition can either be automated, when it includes the interaction with external applications, even requiring the user participation, or non-automated. Automated activities contain hardcoded functionality and require programming activities to implement them.

Figure 1 shows business process management stakeholders interacting through a modeling and execution environment. The stakeholders can play three different roles: executor, modeler and developer. The executor is able to conduct business process execution either by making use of specified activities or create generic activity instances whenever the specified ones cannot fulfill the current execution situation. The modeler is capable of changing the business process model, specifying new non-automated activities. The developer may rely on these non-automated activities and automate them by coding the interaction with external systems. Note that executors, modelers and developers are roles that can be played by the same person.

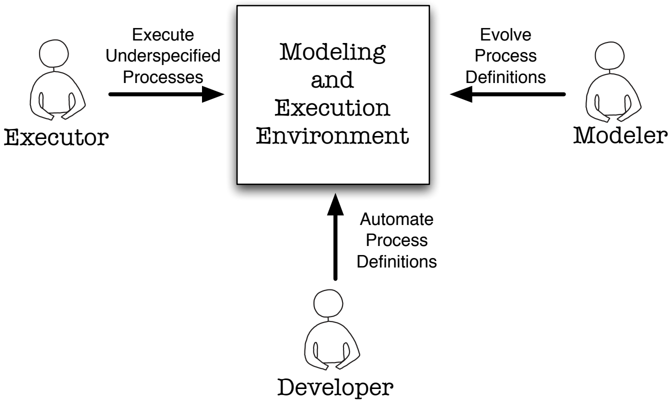


Fig. 1. Modeling and Execution Environment

When executors create generic activity instances they are contributing to the business process model following a case-by-case approach. The generic activity instances capture business process exceptions, allowing the process instance adaptation without requiring all possible situations to be specified in the process model. Moreover, process instance adaptation occurs in the context of process execution, where generic activity instances are integrated with instances of specified activities. Afterwards, executors can tag generic activities and participate in the creation of ontologies for the business process. Following a folksonomy approach, executors tag activities, share their tags, and search for activities based on these tags.[Marlow *et al.* , 2006; Wal, 2007; Wu *et al.* , 2006]

Modelers analyze the set of generic activity instances with its associated folksonomy, and generalize the exceptions over the existing business process model, synthesizing a new version of the model. Once integrated in the model, modeler's suggestions enrich the set of specified activities although in a non-automated form. Afterwards, developers rely on such suggestions to automate the non-automated activities. Developers' decision on which non-automated activities to implement is driven by cost-effectiveness. Cost-effectiveness can be determined by the frequency of activity occurrence in process instances or by the time consumption in activity execution.

Model evolution is a concern of AGILIPO tools. Suggestions are synthesized by modelers following a wiki-like approach[Cunningham, 2002], where new suggestions leverage on previous ones, thus creating new revisions of the model. In this way, AGILIPO intends to foster a knowledge creation process, organically and incrementally (Wikipedia-like)[Riehle, 2008], where contributors are motivated to participate in the modeling of an incomplete process by reading contributions of others and continuously adding their own knowledge[Garud *et al.* , 2008].

As an example, consider an online bookstore and the *Selling* process which has three specified activities: *AddBookToOrder* followed by either *PayWithCheck* or *PayWithCreditCard*. However, as a client goes directly to the physical store and wishes to pay with cash, there is no activity that will cover such situation. The employee may then create an instance of a generic activity and associate it to the current instance of the *Selling* process. Afterwards, the employee needs to assign this generic activity instance to a supervisor because she does not have enough authority to receive the money herself. Therefore, the employee, instead of executing this generic activity instance, addresses it to its supervisor. The supervisor is then able to execute the generic activity instance created by the employee and finish 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 paying with cash exceptional situation. As can be seen, AGILIPO empowers people to perform business processes according to their tacit knowledge and allows responsibility delegation based on the roles played by the organization members: it is up to the employee to know that in this exceptional situation only the supervisor can receive the money.

Both, employee and supervisor can tag the generic activity instance with keywords like for example *Pay*, *Books*, *Money* and *Cash*. That way, executors that in the future get caught in the middle of such exceptional case, could easily find similar occurrences while searching for those tags and easily make use of the same activity instance structure. Moreover, in further executions of the business process, a generic activity instance occurs and has similar tags, then a modeler can decide to specify the *PayWithCash* activity in the model as a non-automated activity. This would create a model evolution such that the employee does not need to search for similar exceptions: the exceptional behavior becomes a suggestion. Finally, the non-automated suggested activity *PayWithCash* could be hardcoded into the application model by a developer automating for example the delegation procedure.

In synthesis, AGILIPO tools support the modeling and execution of business processes, integrate automated and non-automated parts of the process, support both execution and modeling of exceptional behavior, and enforce a continuous knowledge creation process around incompletely defined and understood processes.

5 Social Software Features

AGILIPO strategy for business process modeling is similar to wikipedia's strategy for knowledge gathering [Spek *et al.* , 2006], blurring the distinction between consumers and producers of information. It emphasizes the synthesis of the different suggestions to the business process model through collaborative participation.

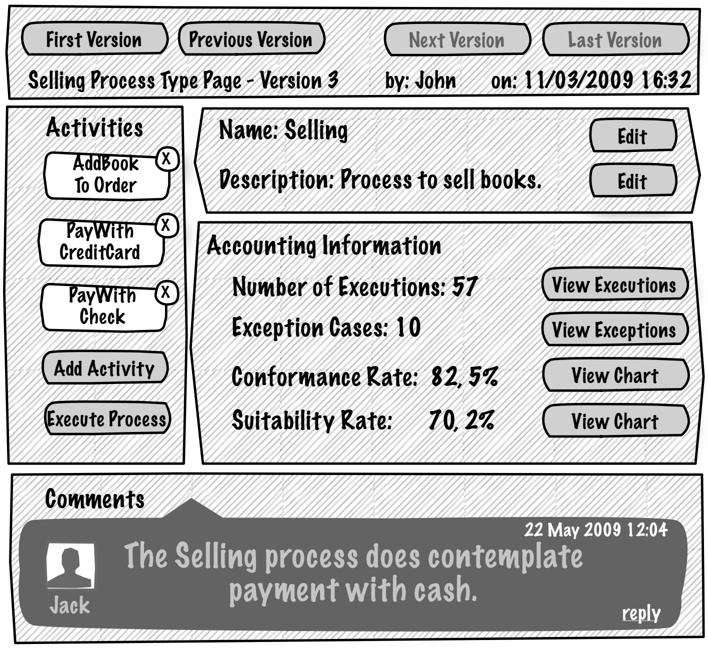
To foster collaboration among executors, modelers and developers, social-software features are used to promote communication:

– **Tagging** - Create folksonomies around generic instances in order to add semantic value to their content and foster business process model evolution.
– **Versioning** - The AGILIPO model is presented in versioned wiki-like pages, keeping track of all suggestions made by modelers and enforcing suggestion synthesis.
– **Comments** - Comments can be used to allow discussion when modelers do not agree on business process model evolution and also to justify execution of generic behavior.
– **Ratings** - Ratings can be used to gather executor's quantitative data about the suitability of the business process model for the particular business process instance she is executing.

AGILIPO business process tools use two sorts of human interaction interfaces: type interfaces and instance interfaces. A type interface provides features to manipulate the business process model while execution is done at an instance interface. For instance, we can have an interface that allows us to make suggestions on the *Selling* process specification and another that allows us to execute a particular selling case, *sell : Selling*.

These interfaces include social software features to foster the bottom-up design of business processes. Figure 2 shows an example of a process type interface for the *Selling* business process presented in Section 4.

The *Selling* business process is on its third version and the last suggestion was created by John. A new version is created whenever a modeler changes



Selling Process Type Page - Version 3 by: John on: 11/03/2009 16:32

Activities

Name: Selling

Description: Process to sell books.

Accounting Information

Number of Executions: 57

Exception Cases: 10

Conformance Rate: 82,5%

Suitability Rate: 70,2%

Comments

22 May 2009 12:04

Jack

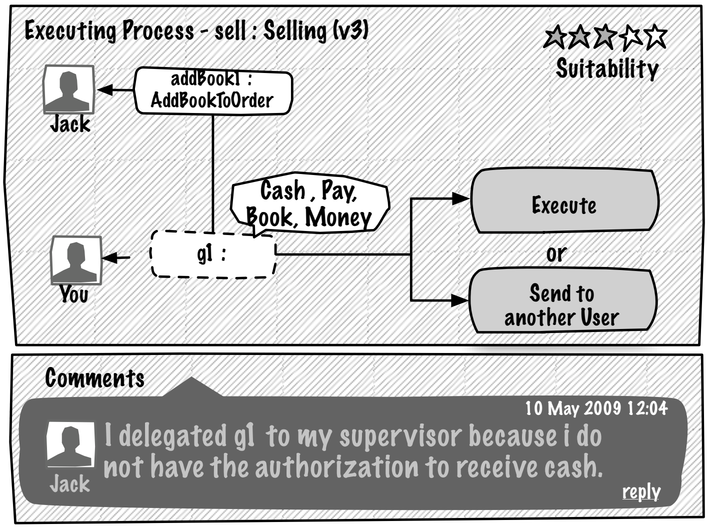
The Selling process does contemplate payment with cash.

Fig. 2. Process Type Interface Example

the process's name or description, or when he deletes, updates or creates an activity type. This will result in a new version of the business process, being possible to navigate between versions. Jack wrote a comment about the lack of the possibility to pay with cash.

The type interface contains accounting information about the business process instances. In this case it is shown the number of instances and exception cases as well as the conformance and suitability rates. The conformance rate is automatically calculated matching the business process definition with the structure of its executed instances. This calculation is based on data mining techniques. The suitability rate represents users satisfaction with the business process definition when they are executing its instances.

Figure 3 shows the execution interface of a *Selling* process instance, which is associated with version 3 of the *Selling* process type. The interface shows the log of executed activities and their executors. It also prompts the executor with the possible next actions. In the shown case, the executor had just created a generic activity instance for payment with cash and tagged it with keywords *Cash*, *Pay*, *Book* and *Money*. The user is empowered to decide whether she receives the payment (Execute) or delegates it to her supervisor (Send to another User). On the top right corner of the interface the executor can rate her satisfaction with the execution of the process instance.



10 May 2009 12:04

Jack

I delegated g1 to my supervisor because i do not have the authorization to receive cash.

Fig. 3. Process Instance Page Example

6 Research Issues

The AGILE BPM approach is rooted in AGILIPO tools, which embed social software features within business processes modeling and execution functionalities. There are several open research problems, related with the design and implementation of the business tool and with the methodological impact of its use that should be addressed:

- What should be the AGILE BPM tool implementation model?
- How should be the wiki-like interface of the AGILE BPM tool?
- What should be the methodology of AGILE BPM?

6.1 AGILIPO Model

AGILIPO Model is based on business process models whose incompleteness is tackled by simultaneously allowing new model suggestions and their respective execution. Such suggestion making reflects flexibility requirements around the implementation model while using social software features to smoothly synthesize the suggestions. AGILIPO tools should be built on a kernel containing an integrated model for both automated business processes and suggestions. This model should uniformly support several degrees of incompleteness and model evolution. Several questions arise.

How does the combined model integrates specific business activities with generic activities? How should we define such generic activities (pre-conditions and execution procedures)? How does it enable the modeling and execution of incomplete business processes? How does it support model versioning? How do we migrate process instance between different versions of the same process type? How can instances be promoted to types fostering the bottom-up design of business processes?

When deciding about the next activity to execute, the tool may advise the executor based on the execution context and on similar past executions. What are the best heuristics to generate recommendations?

The integrated model should support the execution of unplanned exceptions, which include facilities like activity rollback and dynamic change of flow of control. How can we bypass a business process definition? And, how can we return to the pre-specified behavior afterwards?

6.2 AGILIPO Collaborative Interface

AGILIPO Collaborative Interface should empower executors, modelers and developers to simultaneously model and execute business processes. Such collaborative interface should embed social software features. To apply the collaborative interface, upon tools adopting this AGILIPO approach, some questions have to be considered.

Should the suggestion interface be a context sensitive user interface? For instance, when in the context of a particular process instance it shows which generic activities occurred in similar processes, fostering reuse and classification of generic activity instantiations.

How do we seamlessly present automated and non-automated parts of the business process and, at the same time, make them distinguishable in the interface? How do we present suggestions on the automated part of a business process but that do not override the implementation? How do we integrate execution and modeling interfaces?

An important concept to support would be the sense of “neighbourhood”, that is, who are the other users that also have a “proximity” relationship with the context one is visualizing or editing. Who are the most frequent readers? Who are the most frequent editors? Who gives the most comments? What other models do these “neighbours” keep close?

6.3 AGILIPO Methodology

AGILIPO Methodology should foster the collaborative and incremental design and implementation of business processes. To achieve this, we need to address many research questions.

We all know the success of Wikipedia, a collaboratively created encyclopedia, owned by no one and authored by tens of thousands of enthusiasts [Tapscott & Williams, 2006]. Can we downscale the Wikipedia approach to the context of AGILIPO, which is targeted for organizations, and where the number of contributors is very limited? Can some Wikipedia policies, such as consensus, administration, and dispute be used for AGILIPO? Do we need privileges within the AGILIPO approach to moderate dispute? Will these privileges create a distinction among executors, modelers and developers?

Bottom-up and top-down approaches are two different schools of thoughts within the BPM field [Dean *et al.* , 2000]. Each one presents its advantages and disadvantages. The research question which is raised here, is on whether the AGILIPO methodology will enforce the bottom-up business process approach, which is naturally embedded within its concept, or whether there is a need to come up with a hybrid approach, that combines the best practices of both bottom-up and top down approaches, without actually loosing the reality of the working with operations processes?

Process incompleteness is a fundamental aspect of the AGILIPO concept, but this raises a question about the degree of process completeness. When can we decide that the incomplete process is ready for execution? Is the degree of completeness the same for all types of processes? What type of organization design can this fit?

7 Conclusions

Agility is being recognized as a crucial new quality for future BPM approaches [Dreiling, 2009]. In this paper we propose a novel approach for agile BPM based on the embedding of social software features into the business process modeling and execution tools. The distinctive feature of these tools promotes process modeling as a continuous activity that is intertwined with process execution activities, fostering a knowledge creation process that blurs the separation between users and designers of business processes.

The use of AGILIPO tools will impact in the AS-IS/TO-BE paradigm: the AS-IS model is given by the executing business activities while the TO-BE model is given by the incremental changes proposed to the AS-IS model. As soon as incremental changes are implemented, it is not possible to distinguish the TO-BE model from the AS-IS. So, the steps of the AS-IS/TO-BE cycle are unified in a single short duration step where the business processes are perceived AS-Executing and become as incrementally TO-Extend.

To accomplish the AGILIPO vision we identified the need to have an implementation model that smoothly integrates business process features with social software features and a user interface that preserves the wiki-like usability for business process modeling. Moreover, it is necessary to investigate whether the analogy with Wikipedia for knowledge creation downscales in the context of smaller organizations.

References

Beck, Kent, Beedle, Mike, van Bennekum, Arie, Cockburn, Alistair, Cunningham, Ward, Fowler, Martin, Grenning, James, Highsmith, Jim, Hunt, Andrew, Jeffries, Ron, Kern, Jon, Marick, Brian, Martin, Robert C., Mellor, Steve, Schwaber, Ken, Sutherland, Jeff, & Thomas, Dave. 2001. *AgileManifesto*. http://www.agilemanifesto.org (accessed June 1, 2009).

Cumberlidge, Matt. 2007. *Business Process Management with Jboss Jbpm.* City: Packt Publishing.

Cunningham, W. 2002. *"What is a Wiki". WikiWikiWeb.* http://www.wiki.org/wiki.cgi?WhatIsWiki (accessed June 1, 2009).

Dean, D.L., Orwig, R. E., & Vogel, D.R. 2000. Facilitation Methods for Collaborative Modeling Tools. *Group Decision and Negotiation.*

Dreiling, A. 2009. Business Process Management and Semantic Interoperability - Challenges Ahead. *Handbook on Business Process Management - Springer.*

Garud, Raghu, Jain, Sanjay, & Tuertscher, Philipp. 2008. Incomplete by Design and Designing for Incompleteness. *Organization Studies - SAGE Publications*, **29**(03), 351–371.

Golani, Mati, Gal, Avigdor, & Toch, Eran. 2007. Modeling Alternatives in Exception Executions. *In: Business Process Management Workshops.*

Magalhaes, R., & Rito-Silva, A. 2009. *Organizational Design and Engineering.* Tech. rept. Center for Organizational Design and Engineering.

Marlow, Cameron, Naaman, Mor, Boyd, Danah, & Davis, Marc. 2006. HT06, tagging paper, taxonomy, Flickr, academic article, to read. *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, 31–40.

Morgan, Tony. 2002. *Business Rules and Information Systems: Aligning It with Business Goals.* Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Prahalad, C. K., & Krishnan, M. S. 2008. The New Age of Innovation: Driving Cocreated Value Through Global Networks. *McGraw-Hill.*

Riehle, D. 2008. How and Why Wikipedia Works: An Interview with Angela Beesley, Elisabeth Bauer, and Kizu Naoko. *Workshop Wikisym 2008.*

Russell, Nick, van der Aalst, Wil M. P., & ter Hofstede, Arthur H.M. 2006. Exception Handling Patterns in Process-Aware Information Systems. *BPM Center Report BPM-06-04.*

Spek, Sander, Postma, Eric O., & van den Herik, H. Jaap. 2006. *Wikipedia: organisation from a bottom-up approach.*

Tapscott, Don, & Williams, Anthony D. 2006. *Wikinomics: How Mass Collaboration Changes Everything.* Portfolio Hardcover.

Wal, Thomas Vander. 2007. *Folksonomy.* http://vanderwal.net/folksonomy.html (accessed June 1, 2009).

Weske, Mathias. 2007. *Business Process Management: Concepts, Languages, Architectures.* First edn. Springer Verlag.

Wu, Harris, Zubair, Mohammad, & Maly, Kurt. 2006. Harvesting Social Knowledge from Folksonomies. *HYPERTEXT '06: Proceedings of the seventeenth conference on Hypertext and hypermedia*, 111–114.