

BPM and Social Software

Rainer Schmidt¹ and Selmin Nurcan^{2,3}

¹ HTW-Aalen, 73430 Aalen, Germany

Rainer.Schmidt@htw-aalen.de

<http://www.informatik.htw-aalen.de/Schmidt>

² University Paris 1 Panthéon Sorbonne,

Centre de Recherche en Informatique, France

³ IAE de Paris Sorbonne Graduate Business School

Abstract. Social software is software that supports the interaction of human beings and production of artifacts by combining the input from independent contributors without predetermining they way how to do this. Social software enhances business processes by improving the exchange of knowledge and information, to speed up decisions, etc. However, social software can also be used to overcome deficiencies of classic BPM approaches. During the design phase of the BPM life-cycle, social software better integrates the needs of all stakeholders in a more complete way. Using the aggregation mechanisms of social software, constraints for implementation and deployment are captured in finer detail. The aggregation and fusioning of knowledge to cope with incidents is also facilitated by social software. During the evaluation and improve phase, social software highly enhances the collection of suggestions for improvements because each collection can be instantly evaluated by all stakeholders.

1 Introduction

Social software is a new paradigm that is spreading quickly in society, organizations and economics. It supports social interaction and social production. Social interaction is the interaction of non-predetermined individuals. Social production is the creation of artifacts, by combining the input from independent contributors without predetermining the way to do this. Instead, the interaction of the contributors controls the creative process. The artifacts may be content or context information concerning artifacts or physical objects. One could also say that social software is software that supports the production of digital goods such as content, knowledge, software etc. by combining the contributions of individuals that does not necessarily know each other and are a-priori not organized in a hierarchy. Social software follows a more egalitarian and meritocratic approach compared to traditional approaches where the enterprise senior management and its representatives determine the role of the software user. Thus, trust and reputation play a crucial role in the use of social software instead of authority granted by the top management. Based on trust and reputation, hierarchic structures can be avoided nearly completely and self-organizations can be used more widely. If

there are rules and procedures to structure content and organize the cooperation between the users, they are not imposed by a management or authority. Social software is often used without remuneration of the contributors. Instead, the contributors participate for recognition or receive remuneration indirectly by using their recognition to acquire orders etc.

Social software offers new possibilities to enhance business processes by improving the exchange of knowledge and information, to speed up decisions, etc. It facilitates new communication patterns between customers and the enterprise: for example, the communication with the customer becomes increasingly a bi-directional communication with the customer and among the customers. For example, they integrate their customers into product development by using blogs to capture ideas for new products and features. Therefore, more and more enterprises regard social software and social production as a means for further improvement of their business processes and business models. Thus, they integrate their customers into product development by using blogs to capture ideas for new products and features.

However, there is a second relationship between social software and BPM. Social software may also support business process management itself. That means social software is used for the design, operation, improvement etc. of business processes. Thus, in this case social software does not support a certain activity in a business process but the creation, operation and adaption of the abstract business process. The content transformed using social software is a business process model and related information and not run-time-information as in the case before. There are three issues in the use of BPM where the use of social software could help.

1.1 The Model-Reality Divide

On workshops and conferences such as BPMDS series, BPM 2008 there is a growing concern about the divide between abstract process models and the executed processes, procedures etc. This model-reality divide means, that although BPM models and structures are well designed, they are not used in practice but end up in the filing cabinet. Furthermore, the employees do not accept the models created by the BPM-department or consultants. Instead, the employees do a lip service during the process rollout, but quickly ignore the processes as soon as the project is finished. Consequently, after short time, the modeled and the executed (real) processes fall apart.

1.2 Lack of Information Fusioning

One cause for the model-reality divide is that process users are not properly involved into Business Process Management. They are only “consumers” who are forced to accept the processes created for them. Furthermore, the unification of terms and the integration of knowledge is not done on a peer-to-peer basis instead, a “par ordre de Mufti” approach is applied. That means the terms and concepts are simply imposed on the employees.

1.3 Information Pass-On Threshold and Lost Innovation

However, even if the processes are rolled-out successfully, a substantial potential for process improvement is not used due the information pass-on threshold. The information pass-on threshold denotes, that ideas for improvement are not passed on to the responsible because this creates too much effort for the user (“Why shall I write a memo, letter, etc. ”), the further processing is not transparent (“What will happen with my suggestion ..”) or the success is considered as improbable (“Will not succeed anyway ..”). The entering of information is strongly regulated; the process to submit changes is too restrictive. In addition, formal process models create a threshold for participation, because users without modeling knowledge cannot bring in their ideas. Therefore, important and valuable information is lost and improvements remain undone. As a consequence, the implemented process differs more and more from the best practice and the employees tend to do their own “private” process which contains the optimizations that they have regarded too difficult to integrate in the “official” process. Thus again a model-reality divide is generated.

This introductory paper will investigate the relationship between BPM and social software especially with respect to the support of BPM by social software and proceed as follows. First, the roots of social software are illuminated to show, that the success of social software shown by examples such wikipedia.org is not by accident but in accord with important streams of research in economics, sociology and marketing. Social software is not a new approach; there are predecessors and approaches with similar intentions but using other concepts that are shown in the following chapter. In addition, the artifacts created by social software are investigated. Then an evaluation of social software is made and the types of social software are differentiated. Based on this foundation, the relationship of BPM and social software is analyzed for every phase of the BPM life cycle. Finally success factors for the use of social software are given.

2 The Roots of Social Software

The success of enterprises based on social software such as facebook or MySpace obscures, that there are important streams of scientific research leading to social software. Thus, the success of social software is not only an empirical phenomenon nor a pragmatical requirement, but can also be well explained by science. Four roots for the success of social software can be identified, as shown below:

2.1 Weak and Strong Ties - Granovetter

Social software supports the creation of weak ties. Weak ties are a concept identified by Granovetter [3] [4] . They are connections between individuals enabling them to reach information etc. not accessible via strong ties imposed by corporate hierarchy or by team membership. Weak ties are crucial to improve enterprise agility and innovation because they help to find information not available in the team or department of the individual. The concept of weak ties is

based on the criticism of traditional paradigms for productions such as Fordism (mass-production) and Taylorism (division of labor). These paradigms strongly emphasize strong ties and thus impair the capability to innovate.

2.2 Wisdom of the Crowds - Surowiecki

Social software is a mechanism that supports the ideas, which Surowiecki collected under the title “wisdom of the crowds” [6]. They say, that for many decision and planning problems which do not allow to “calculate” the optimal solution, a near-optimal solution can be found by combining as many inputs as possible. There may be some experts that deliver a better solution. However, only a few percentages of the experts are better and in advance, it is not possible to identify those experts. A prominent example is the stock exchange. Only few investment funds perform better than the indexes such as Dow Jones and it is not possible to predict which ones. The idea of the wisdom of the crowds is also embedded in democracy, the foundation of western societies. The policy “one man - one vote” is expression of the trust in decisions met by combining many different votes and not from a few experts.

2.3 Social Production - Benkler, Tapscott

The isolated creation of proprietary products is questioned more and more by authors e.g. Benkler [1], Tapscott [7]. These authors say, that you must open your company to capture new ideas from outside and you have to cooperate with many different people to combine the best thoughts and create competitive product. Social software offers many possibilities to interact with the customer and partners. For example, blogs allow collecting ideas for product improvement quite easily.

2.4 Service-Dominant Logic

Service-Dominant Logic [8] is a highly successful approach in marketing which says, that the traditional, goods-oriented approach for marketing has to be replaced by a service-oriented one. Service-Dominant logic postulates, that the customer does not want a product but the service provided by the product [8]. Furthermore, it is necessary to interact with the customer to provide the service. The customer is not a consumer of value but a co-creator of value. To achieve this vision of collaborative marketing, social software is an important means.

3 Social Software and Its Relation to Other Approaches

The concept of social software is in relation to a number of approaches. First, there are predecessor technologies, which implemented one or several core concepts of social software such as the continuous fusioning of information originating from different self-organizing users. Furthermore, there are approaches

such as groupware and knowledge management which pursue similar intentions but use different concepts. Social software is often implemented using Web 2.0 technologies such as Ajax etc. that allow to deploy easily application functionality in the WWW. Social software supports the concept of Enterprise 2.0 and implements the concept of a Social Web.

4 Principles of Social Software

The principles driving the creation of content and context constitute an important difference of social software when compared to other approaches. The most fundamental difference is, that there is no pre-defined development process as found in software engineering e.g. Instead creative processes, which execution support is possible thanks to social softwares, can be described by the following principles.

- Self-organization and bottom-up organization
Information is not classified, structured, organized by a specialist but by the community of users. There are no pre-determined experts that impose a structure on the content. Instead, all users develop the structures interactively.
- Egalitarian
All users have (nearly) the same rights to enter content. There is no separation between the contributor and the consumer of content and context information. In addition, the low input effort lowers the threshold to contribute. No knowledge impedes you to become an author.
- Continuous and immediate fusioning and aggregation
Content from different sources is fused and aggregated continuously and becomes immediately visible and effective.
- Continuous and recursive assessment
Contributions are under continuous assessment of all users. Flaws detected can be corrected immediately and need not to be delegated to the author, administrator. Assessments can be recursive. Example “do you consider the assessment valuable”
- Both content and context are considered valuable
Knowledge is not only in content but also in the context of usage of the content, for example the connections between the content items. Other examples are tags, links, bookmarks, etc.

5 A Classification of Social Software

Social software can be classified in two dimensions. First it can be classified according to the artifact created, second it can be differentiated whether it fuses or aggregates the artifacts.

Social software is used to create two kinds of artifacts: content and context. Content may be of different types such as text or multi-media. Context can be

further differentiated into three sub-types: Annotation, reputation and social links. Annotation is information that helps to understand, find, and evaluate objects. These objects may be content in the social software or real objects. Reputation is a substitute for trust in social software. Because most users of social software do not know each other, it is necessary to provide reputation information. The third subtype of context in social software is social links. They provide information about connection between human beings.

It is important to note, that the creation of open source software and the formation of prices show a tight relationship with the ideas of social software. Open Source Software such as Linux is developed by a large number of independent developers. The association of price formation to social software may surprise at first sight. However, the formation of a price in a market is nothing else than putting together the offers and bids from many individuals to create a new piece of information. Based on these considerations we can identify a spectrum based on the complexity of artifacts created. At the left hand side is the formation of prices in a market with a very low complexity of the artifact created: the price formed is a number. On the right hand, side there is the creation of open source software, a very complex artifact. Between them are content and context provided by social software.

The second dimension for the classification of social software differentiates the integration of the inputs from the contributors. Some types of social software fusion the information contributed in a way that a new artifact is created and the contributions merge with one another. An example is the fusioning of information in a wiki. Other types of social software only aggregate the contributions. For example, a blog only juxtaposes information and links it.

Based on the considerations above, some popular types of social software shall be discussed and classified. However, it should be noted, that popular implementations often mix different types such as the use of tagging in wikis.

5.1 Wikis

Wikis are used to create both content and context information. Authors create text and multimedia and add links to create the context. The core of a wiki is a collaborative editing mechanism combined with an optimistic locking and access mechanism. The access of many wikis is completely unrestricted. This appears incautious, but is not so relevant, because wikis have an elaborate versioning mechanism, which makes every change from every user distinguishable and resettable. Changes may be easily detected by comparing two versions. Thus, malicious users do not have the chance to create a huge damage. The versioning mechanism also allows resigning on locking and check-in / check-out mechanisms to coordinate the user's changes.

In a wiki, not only textual content is generated but also the context related information is provided using links and categories. Thus, relationships between information may be easily edited collaboratively. Important extensions of wikis are Semantic Media Wikis [9] [5]. They allow creating typed links.

5.2 Blogs

Contrary to wikis, blogs do not allow to fusion content. Instead, the contributions of the user stay as separate entities. However, the blog entries can be used to annotate other entries or content. Therefore, blogs are primarily used as interaction mechanisms documenting the thread of communication.

5.3 Tagging and Social Bookmarking

Tagging consists to provide context to related information to objects by the association of tags. The tags may be freely chosen and are not part of hierarchy. Therefore tagging creates a flat world from an uncontrolled vocabulary, contrary to hierarchical classification systems, taxonomies etc. Objects are not exactly classified by terms from a controlled vocabulary, but by a cloud of tags, the tag cloud. A tag cloud is the weighted set of tags associated with an object. The tag cloud is also used to indicate the usage of tags for tagging and searching in a tagging environment.

Due to the lack of a controlled vocabulary and a hierarchy of terms, there is no control of synonyms and homonyms. On the contrary, this fuzziness implies also an extensibility that allows catching semantic differences that would have been ignored by a pre-defined vocabulary. Social bookmarking is the collaborative collection of bookmarks. It often overlaps with tagging because the collected bookmarks are organized using a tagging mechanism. Del.icio.us for example allows users to share their tagged bookmarks.

5.4 Recommender and Reputation Systems

The principles of social software are more and more applied in recommender systems. Both aggregation and fusioning can be found. Many recommender systems for books, hotels etc. aggregate individual reviews. Additionally they compose an evaluation from a multitude of contributions from users. Evaluations may be created from explicit statements from users or by observation of their behavior. For example, tripadvisor.com collects the explicit evaluations of hotels. Amazon.com shows the percentage of visitors who bought a book.

However, social recommendation systems are susceptible for camouflage. Therefore, they are combined with reputations systems that allow weighting a user's contribution with its reputation. E. g., reviewers at amazon.com are annotated with information expressing their reputation, for example the number of their reviews. Reputation system may also be used on their own. For example, the rating mechanism of eBay or the evaluation mechanism of the amazon.com marketplace determines the reputation of a seller or buyer by using the independent contributions of their business partners.

5.5 Social Links

Social links differ from links in wiki, because they can only be defined by one person. Only the user himself may declare friendship etc. with another user.

Social links are used in social platforms such as facebook.com, xing.com. Their main purpose is to create contacts between the participants. These platforms do primarily aggregation, but functions like “friends of my friends” are a kind of fusioning mechanism.

6 BPM Support Offered by Social Software

Based on the considerations above, a lifecycle perspective will be used to analyze the possibilities provided by social softwares to support BPM [2]. It shall also be investigated, which types of content or context related knowledge may be created by social software during the BPM life-cycle and whether content and context are fusioned or aggregated.

6.1 Design

During design phase, fusioning capabilities of social software can be used to create a specification that better integrates the needs of all stakeholders. The requirements can be collected more completely, by lowering the threshold to contribute to the specification. Furthermore, social software facilitates to create a real common understanding of terms and definitions. By this means, a truly participative modeling approach (i.e. a co-production) is implemented. Social software can also help to create reference models more easily. Now, a lot of edit - integrate cycles are necessary to create a reference model from the contributions. However, using social software, this integration may be much easier due to its fusioning capabilities.

6.2 Implement and Deploy

The success of the implement and deploy phase is highly dependent from a very broad input of possible constraints. Using the aggregation mechanisms of social software, constraints for implementation and deployment are captured in finer detail. Especially during deployment, the broad collection of relevant issues and distribution of planning data is of high importance for project success. Social software may help to collect issues and to broadcast all relevant planning information.

6.3 Operate

During the operation of the business process, interruptions by incidents have to be reduced as much as possible. Crucial for this task, is the aggregation and fusioning of knowledge to cope with these incidents. Social software highly facilitates this task.

6.4 Evaluate and Improve

Social software highly enhances the collection of suggestions for improvements because each collection can be instantly evaluated by all stakeholders. Furthermore

it seems much easier to handle priorities while the priorities of the stakeholders can be fused using social software.

7 Summary

Social software is software that supports the interaction of human beings and production of artifacts by combining the input from independent contributors without predetermining their way to do this. The artifacts created may be content or context information concerning abstract or physical objects. Social software is based on the principle of self-organization that also applies to the structuring of information. Information is not classified, structured, organized by a specialist but by the community of users. It does not differentiate between the contributor and the consumer of content and context information. All contributions are fused and aggregated continuously and become immediately visible and effective. Thus, a continuous assessment is possible.

Social software enhances business processes by improving the exchange of knowledge and information, to speed up decisions, etc. Thus, new communication patterns between customers and the enterprise appeared. However, Social software can also be used to overcome deficiencies of classic BPM approaches. It can at least narrow the model-reality divide which manifests in the falling apart of process models and real occurrences of these models. Social software also offers a better information fusion by lowering the threshold to contribute. During design, social software allows to better integrate the needs of all stakeholders in a more complete way. Using the aggregation mechanisms of social software, constraints for implementation and deployment are captured in finer detail. The aggregation and fusion of knowledge to cope with incidents are also facilitated by social software. During the evaluate and improve phase, social software highly enhances the collection of suggestions for improvements because each collection can be instantly evaluated by all stakeholders.

In summary, social software allows to integrate nearly all users into the design and implementation of business processes. Social software facilitates the administration of information that contains multiple perspectives, which cannot be linearized, homogenized etc. Information can be approached from different perspectives. Links allow directing from different sources to the information. Tagging allows associating meta-info from different perspectives. Thus the divide between abstract process models, lifecycles, evaluations and the executed processes, procedures etc. can be narrowed or even completely avoided. The lack of formal barriers also tears down psychological barriers. Excuses are no longer possible due to a very low entrance barrier. No delegation can explain the omitting of changes. Instead, due to the immediate effects of employee action, their involvement and commitment are increased. However, there are also disadvantages, which originate primarily from the lack of hierarchy. Often overheads for the self-organization of the contributors increase.

References

1. Benkler, Y.: *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press (2006)
2. The 9th Workshop on Business Process Modeling, Development, and Support (BP-MDS 2008) in conjunction with CAISE 2009. Business Process Life-Cycle: Design, Deployment, Operation and Evaluation, 16-17 June, Montpellier, France, <http://lamswww.epfl.ch/conference/bpmds08>
3. Granovetter, M.S.: The Strength of Weak Ties. *American Journal of Sociology* 78, S.1360 (1973)
4. Granovetter, M.: The strength of weak ties: A network theory revisited. *Sociological Theory* 1, S.201–S.233 (1983)
5. Krötzsch, M.u.a.: Semantic Wikipedia. In: *Web Semantics: Science, Services and Agents on the World Wide Web* (2007)
6. Surowiecki, J.: *The Wisdom of Crowds: Why the Many Are Smarter Than the Few and How Collective Wisdom Shapes Business, Economies, Societies and Nations*, Doubleday (2004)
7. Tapscott, D., Williams, A.D.: *Wikinomics: How Mass Collaboration Changes Everything*, Portfolio (2006)
8. Vargo, S., Lusch, R.: Service-dominant logic: continuing the evolution. *Journal of the Academy of Marketing Science* 36, S.1–S.10 (2008)
9. Völkel, M.u.a.: Semantic Wikipedia. In: *Proceedings of the 15th international conference on World Wide Web*, pp. S.585–S.594 (2006)