BPMN and Design Patterns for Engineering Social BPM Solutions

Marco Brambilla, Piero Fraternali, and Carmen Vaca

Politecnico di Milano, Piazza L. da Vinci 32, Milano, Italy,
name.surname@polimi.it

Summary. The integration of social software and BPM can help organizations harness the value of informal relationships and weak ties, without compromising the consolidated business practices embedded in conventional BPM solutions. This paper presents a process design methodology, supported by a tool suite, for addressing the extension of business processes with social features. The social process design exploits an extension of BPMN for capturing social requirements, a gallery of social BPM design patterns that represent reusable solutions to recurrent process socialization requirements, and a model-to-model and mode-to-code transformation technology that automatically produces a process enactment Web application connected with mainstream social platforms.

Key words: Model-Driven Engineering, BPM, BPMN, Social Software, Design Patterns, Generative Development

1 Introduction

Social BPM fuses BPM with social software, with the aim of enhancing performance by means of a controlled participation of external stakeholders to process design and enactment [6, 14, 20].

In classical BPM, processes are defined centrally by the organization and deployed for execution by *internal performers*, i.e., actors formally entitled to execute the activities and directly produce the advancement of a process case. This closed-world approach can be opened with social features at different levels of control [2]: *Participatory Design* opens the *process design* to multiple actors, including end users; the resulting process is then executed in the traditional way; *Participatory enactment* shifts socialization from design to enactment and allows the participation of *internal observers*, i.e., actors known at design time (e.g., internal to the organization) but different from the internal performers formally entitled to activity execution; these subjects can interact with (observe) the process only indirectly, via the intermediation of messages and artifacts; finally, *Social enactment* enlarges even more the process execution, by allowing the participation of *external observers*, i.e., actors not known at process deployment time and dynamically signed-up to the process.

This paper focuses on participatory and social enactment and contributes:

– A summarization of the main factors that drive the socialization of a business process (socialization goals).
– An extension of BPMN 2.0 enabling the specification of social roles, activities, events, and process flows (Social BPMN).
– A gallery of design patterns, expressed in Social BPMN, that represent archetypal solutions to recurrent process socialization problems (social process patterns). Social patterns are referred to the goals they contribute to solve, to support the construction of process models from requirements.
– A technical framework for generating Social BPM applications from specifications encoded in Social BPMN, based on model transformations and on a runtime architecture integrating business process execution and social task enactment, implemented in a commercial tool suite called WebRatio BPM [1].

The paper is organized as follows: Section 2 provides an overview of the approach and summarizes the socialization goals; Section 3 describes the notation extensions to the BPMN language; Section 4 presents the social BPM design patterns; Section 5 provides a complete example of the approach; Section 6 describes the technical framework for process modeling and application generation; Section 7 discusses the related work; and Section 8 concludes.

2 Overview of the Approach

Figure 1 positions the contributions of the paper with respect to the phases the BPM lifecycle.

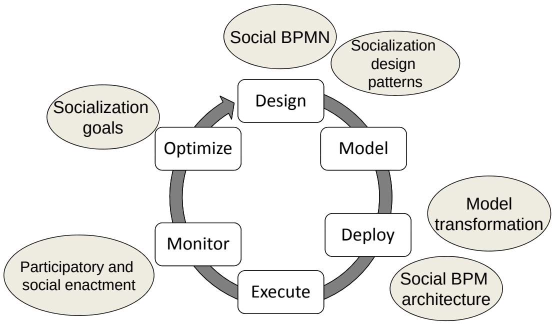


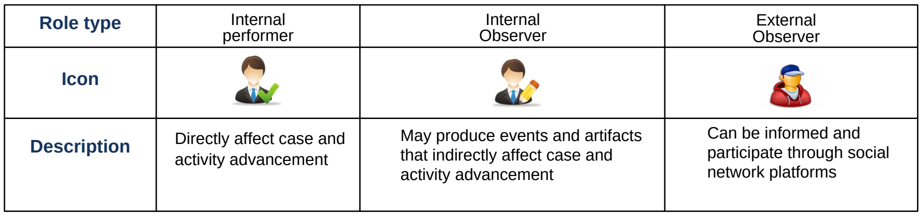
Fig. 1. The BPM lifecycle and the contributions of Social BPM.

The social extension of a business process can be regarded as a specific optimization phase, where the organization seeks efficiency by extending the reach of a business process to a broader class of stakeholders. This general objective articulates into different optimization goals, which constitute the motivation of the process socialization effort:

- *Exploitation of weak ties and implicit knowledge*: the goal is discovering and exploiting informal knowledge and relationships to improve activity execution.
- *Transparency*: the goal is making the decision procedures internal to the process more visible to the affected stakeholders.
- *Participation*: the goal is engaging a broader community to raise the awareness about, or the acceptance of, the process outcome.
- *Activity distribution*: the goal is assigning an activity to a broader set of performers or to find appropriate contributors for its execution.
- *Decision distribution*: the goal is eliciting opinions that contribute to the making of a decision.
- *Social feedback*: the goal is acquiring feedback from a broader set of stakeholders, for process improvement.
- *Knowledge sharing*: the goal is disseminating knowledge in order to improve task execution; at an extreme, this could entail fostering mutual support among users to avoid performing costly activities (e.g., technical support).

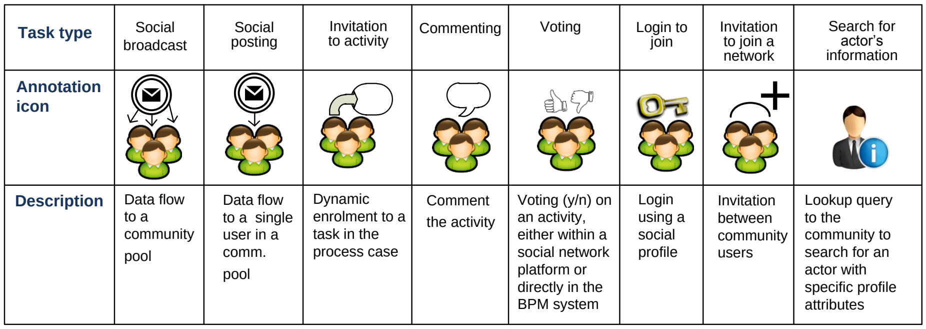
Once the optimization goals are established, they must be incorporated into the process design phase. This poses a linguistic problem (how to express socialization in the process model) and a procedural problem (how to design the process model so that it fulfills the socialization goals). BPMN 2.0 native extension mechanism can be exploited to provide a quite natural answer to the former issue: the main concepts of the language can be stereotyped to convey their social extension. To support the designer in constructing process models that meet the socialization goals, pattern-based design can be exploited [9]. A design pattern is *general reusable solution to a commonly occurring problem*; by identifying social process design patterns and matching them to socialization goals, it is possible to assist developers in the modeling of social BPM solutions that meet their process socialization requirement. Section 4 lists the collection of design patterns that we have derived by an extensive literature review and by the available descriptions of implemented social BPM products and solutions.

When the model of the social process is consolidated, the *deployment* consists of a the technical phase that produces the actual executable version of the social process enactment application. This task is complicated by the need of interacting at runtime with social software to support the social interactions required by the process model; these platforms are available online and can be used as a service in the enactment of the process (e.g., LinkedIn for skill and people search, Doodle for decision distribution, etc.). However, the integration of the BPM runtime to the social services is a nontrivial task, complicated by the absence of an interoperability standard masking the technical details of the APIs of each different platform. To face this problem and support deployment, Section 6 illustrates a technical architecture and development tools that automate the generation of process enactment applications from Social BPMN process models. The architecture and tools use the WebML Domain Specific Language (DSL) [3] and model transformations to first map Social BPMN models into platform-independent WebML application models and then the WebML models into Java components connected to social software APIs.



Role type	Internal performer	Internal Observer	External Observer
Icon			
Description	Directly affect case and activity advancement	May produce events and artifacts that indirectly affect case and activity advancement	Can be informed and participate through social network platforms

Fig. 2. BPMN lanes and pools stereotyped to denote social actors.



Task type	Social broadcast	Social posting	Invitation to activity	Commenting	Voting	Login to join	Invitation to join a network	Search for actor's information
Annotation icon								
Description	Data flow to a community pool	Data flow to a single user in a comm. pool	Dynamic enrolment to a task in the process case	Comment the activity	Voting (y/n) on an activity, either within a social network platform or directly in the BPM system	Login using a social profile	Invitation between community users	Lookup query to the community to search for an actor with specific profile attributes

Fig. 3. BPMN tasks stereotyped to denote social activities.

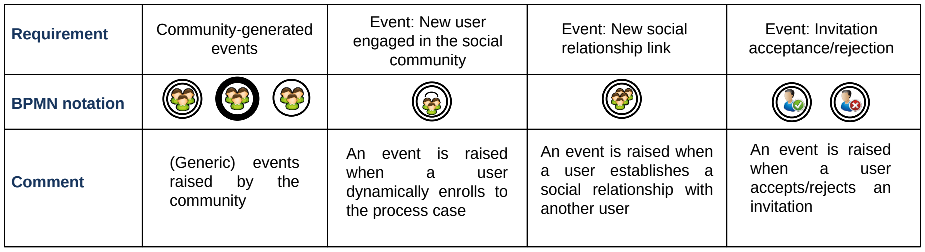
3 Social BPMN extensions

Process design benefits from visual languages that convey the process structure and constraints in an clear way, immediately communicable also to non-technical stakeholders. Social process design should preserve the intuitiveness and expressivity of state-of-the-practice visual languages and possibly be based on standard notations. To this end, social extensions of business processes can be conveyed using the BPMN standard[1] as a linguistic base. BPMN 2.0 incorporates a native extension mechanism that makes the language well suited for the adaptation to special process requirements, like those arising in Social BPM. By enriching the existing BPMN concepts with a social meaning, it is possible to achieve a visual language that is both familiar to BPMN practitioners and possess enough expressive power to convey social design patterns.

The main social extensions refer to the possibility of assigning work to users different from internal performers, and on the special semantics attached to some community-performed activities and events. Figure 2 shows the notation for *social pools*: social users are denoted by a stereotype icon adorning the BPMN pool, so to distinguish internal performers (corresponding to the standard semantics of BPMN pools) and the pools formed by the social communities of internal and external observers.

Social activities are activities executed by multiple actors; they can be represented with the BPMN 2.0 concept of ad hoc parallel task (the social pool

[1] http://www.bpmn.org/



Requirement	Community-generated events	Event: New user engaged in the social community	Event: New social relationship link	Event: Invitation acceptance/rejection
BPMN notation	◎ ◉ ◎	◎	◎	◎ ◎
Comment	(Generic) events raised by the community	An event is raised when a user dynamically enrolls to the process case	An event is raised when a user establishes a social relationship with another user	An event is raised when a user accepts/rejects an invitation

Fig. 4. BPMN event types supporting social interactions.

of figure 8 contains several examples). In particular, *Social tasks* specialize the BPMN task concept to denote a process action with a social semantics: they are denoted by an icon that suggest the social meaning of the task, as exemplified in figure 3: the broadcasting of messages/contents from a task to the entire social network (or a subset thereof), the posting of messages/contents to one member of the network, the invitation of people from the social network to perform a specific task, the invitation to comment or vote on a task or on its outcomes, the login of users in the BPM system using credentials from a social network, and the search for user's skills or reputation within a social network (e.g., for checking recommendations before assigning tasks to users). The control of social tasks exploits standard BPMN sequence flows, either within a social pool or between a social and a normal pool. As shown in figure 4, specialized event types can be used to denote case advancement triggered by social interactions. A generic social event concept represents any kind of occurrence within the social network; this can be specialized to express more detailed event types like: the addition of a new user to the community, the establishment of a new social relationships, the notification of acceptance/rejection of a social request (e.g., for friendship, invitation to groups or applications), and so on. Social pools, actor categories, social tasks and events are the linguistic building blocks for expressing social design patterns, that are archetypal process model fragments representing recurrent process socialization solutions. They are the subject of Section 4.

4 Social Design Patterns

Social design patters are solutions to recurrent scenarios where cooperative tasks are executed using social software. This section illustrates an initial gallery of design patterns, collected by a broad review of social BPM literature and from an ongoing experience of analysis with process owners in companies and public administrations. For brevity, every pattern is introduced by a statement of the problem it addresses, the sub-patterns specializing it or used by it, and a short description explaining the supported social interaction.

Dynamic enrolment: Involving people external to the process. *Sub-patterns*: Open invitation for external observers; Closed invitation for inter-

nal observers, Self-enrolment.
Description: platforms like enterprise and public social networks[2] are exploited for dynamically adding new actors to social activities. Internal/external observers (not a priori registered to the process) are invited through social software to sign into the process. Figure 5 shows the BPMN representation of the open invitation pattern using the annotation icons to denote social activities. Internal performers generate an invitation by sending a message to a social pool, and internal/external observers sign and start contribute to some social task.

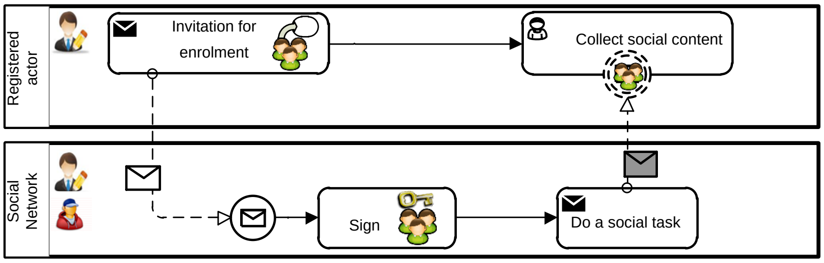


Fig. 5. Dynamic enrolment pattern

Poll: Cooperating to a social decision.
Description: An internal performer publishes to a social platform a question (e.g., an open or closed list of options to choose from). Internal/external observers receive an invitation to participate in the poll [11, 15], with a termination condition (e.g., a deadline). After the termination event, the internal performer collects the contributions and uses them to produce the decision, which can be published back to the social platform. Figure 6 shows the BPMN representation.

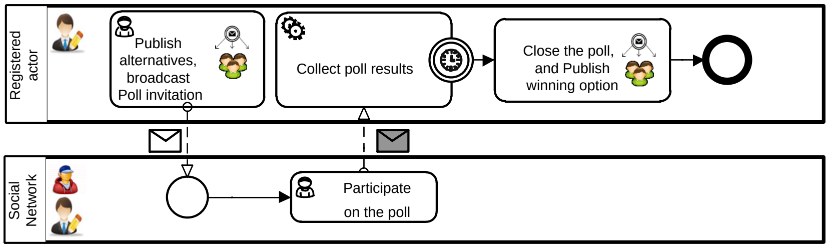

Registered actor
Publish alternatives, broadcast Poll invitation
Collect poll results
Close the poll, and Publish winning option
Social Network
Participate on the poll

Fig. 6. Poll pattern

People/skills search: Finding competencies for an activity.
Description: A social community, inside and/or outside the enterprise, is exploited to find people with required expertise and a choice is made trading

[2] Example of enterprise social network are Yammer and Jive; of public ones are Facebook, LinkedIn and Twitter.

competence and social distance [5, 10]. The process usually starts by publishing a call for people, to which internal/external observers respond. The internal performer selects the right candidate(s) and publishes the final decision.

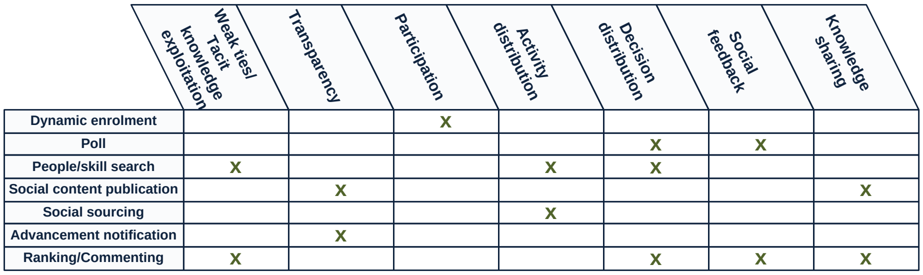
Social publication: Making a process artifact visible to social actors.
Description: Content (e.g., a public directive) is published to internal and external observers, e.g., by posting a document to a social platform [11]. Artifacts contain limited views of the process information. The social task terminates with the production of an advancement event (e.g., a deadline or the achievement of a required number of content views/comments).

Social sourcing: Delegating an activity to social actors.
Sub-patterns: Content creation, Content enrichment
Description: Internal/external observers contribute to the execution of an activity, e.g., by co-authoring socially produced documents [11, 20]. Internal performers publish the description of the work and share a resource link to start contributions. It is possible also to enrich already existing content: creating metadata, tagging people/artifacts, e.g., a user finds a relevant colleague's profile, adds it to a bookmark collection and tag it [7, 8, 19].

Advancement notification: Informing social actors about advancement.
Description: Social contribution to the process can be fostered by delivering timely information on the progress status of activities. Using micro-blogging platforms [18], for example, it is possible to keep the users updated on limited views reflecting activity/case advancement making process execution more transparent. This design pattern lets an internal performer mark an activity as socially notified, so to generate automatic progress messages to selected social networks.

Feebdack Acquiring qualitative/quantitative feedback from social actors.
Description: The internal performers may produce artifacts in the process execution and evaluate them by asking internal/external observers to rate them or to insert comments into the social platform [11].

Figure 7 shows how the defined social patterns address the optimization goals listed in Section 2 and can be used to drive the construction of a social process model after the process improvement analysis phase.



	Weak ties/ Tacit knowledge exploitation	Transparency	Participation	Activity distribution	Decision distribution	Social feedback	Knowledge sharing
Dynamic enrolment			X				
Poll					X	X	
People/skill search	X			X	X		
Social content publication		X					X
Social sourcing				X			
Advancement notification		X					
Ranking/Commenting	X				X	X	X

Fig. 7. Principal goals covered by each social pattern.

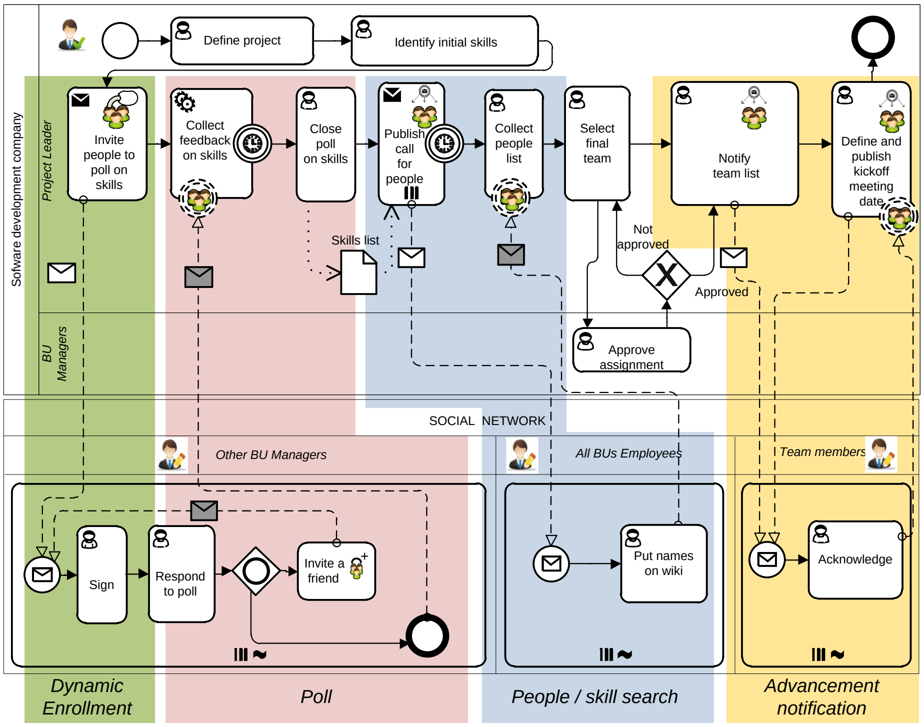


Fig. 8. The socialized Team formation process

5 A Complete Example

To illustrate the pattern-based modeling approach, an example of process socialization is presented. A multinational software firm has formalized the process of team creation for projects requiring software architects, developers, system specialists and domains experts. In the traditional process, a senior manager of the Business Unit (BU) where the project is conducted is appointed as a team leader and s/he constructs the team based on staff availability and personal relationships; however, past experience has demonstrated that a more flexible approach is desirable, not to overlook hidden skills that may not be apparent from the project definition and to cope with projects in sectors new to the responsible BU. To this end, several of the socialization goals introduced in Section 2 are relevant: *participation*, to involve other BUs as advisors in the team formation; *exploitation of weak ties and tacit knowledge*, to make hidden skills surface; *decision distribution*, to exploit knowledge external to the BU in the team building; and *Advancement notification*, to make the process transparent to contributors. The identification of the primary socialization goals allows the process designer to select a candidate set of patterns, using a relevance matrix like the one shown in Figure 7; these patterns form an initial base for process improvement. Figure 8 shows the (simplified) outcome of process re-design, where the Dynamic Enrollment, Poll, People/skill Search and Advancement Notification patterns are used to meet the socialization goals.

In this version, the actors belong to the owner BU pool, denoting internal performers, and to a social pool of internal observers, denoting managers of other BUs and all the employees of the company. The Project Leader starts the process by creating a project description and an initial skill set; then s/he starts a poll on the skill list and submits it to the social pool (this can be realized by a post on the enterprise social network with a link to an external poll service). Other BU managers act as internal observers; they enroll dynamically to the process and propagate the invitation to other relevant colleagues thanks to the friendship mechanism of the enterprise social software. Comments can be added after completing the poll, so that colleagues from other BUs can suggest skills that have not yet been considered. After a fixed time, the Project Leader collects the feedback on the skill's list, closes the poll and publishes a Call for people to all the employees. The call for people is also a socialized activity; employees can suggest other colleagues or can promote themselves by giving information about their professional experience in similar areas. After a period of time, the Project Leader defines the composition of the team, shares it for approval with the managers of the responsible BU that will work in the project, notifies the selected team members, and finally publishes the kickoff meeting date.

6 Automatic Generation of Social BPM Solutions

The proposed framework has been implemented in WebRatio [1], a Model-Driven Web application development tool allowing one to edit BPMN models and automatically transform them into running JEE applications. The code generation exploits an intermediate platform-independent application model, expressed in the WebML language [3], so that application developers can fine-tune the Web application for enacting the process, by enriching the skeleton application model produced automatically from the BPMN process diagram.

A rapid prototyping function applies directly to the social process model and lets a business analyst or a stakeholder: 1) impersonate any actor of the process, at all the levels of social interaction; 2) start/suspend/resume/terminate the process activities; 3) create and inspect project artifacts and parameters, according to the process specification; 4) impersonate external user roles and play social actions. The prototype can be refined at the WebML modeling level by editing the BPMN or WebML models and then re-executing the model transformations, until the resulting application meets the requirements for deployment. Figure 9 provides an overview of the implementation framework. At design time, the analyst creates the BPMN process models. Then, the automatic transformation from BPMN to WebML considers the type of the gateways and of the tasks (User or Service), as well as the information on the control and data flows, to generate a Web application model for process enactment. The Web application model consists of: WebML components expressing the business logic of user-driven and automatic tasks (e.g., tasks performed by Web services) and WebML components expressing the hypertext interface for managing the tasklist and the process execution status. In particular, social BPMN tasks are transformed into WebML application-level patterns, which make use of components for connecting to the social software. Process deployment exploits the transformation from WebML to the Java code, which is already implemented in WebRatio [1] and has been extended to support the social BPM patterns. The visual presentation and the business logic of the application can be customized by extending the components predefined in the WebML language with additional custom components, to obtain any desired behavior. Using this mechanism, a set of new WebML components and transformation rules have been implemented to realize the social BPM patterns and connect the resulting enactment application to the social networking platforms needed for social behavior. The code generated from WebML models is a standard Java application, which can be deployed on any Java application server. Connectivity to the social software is realized by APIs calls to the external platforms, abstracted by means of WebML components. Examples of the implemented connectors include bridges to popular social software like Twitter, Facebook and LinkedIn, wrappers of open authentication systems like OAuth, and connectors to Web utility services like Doodle for polls.

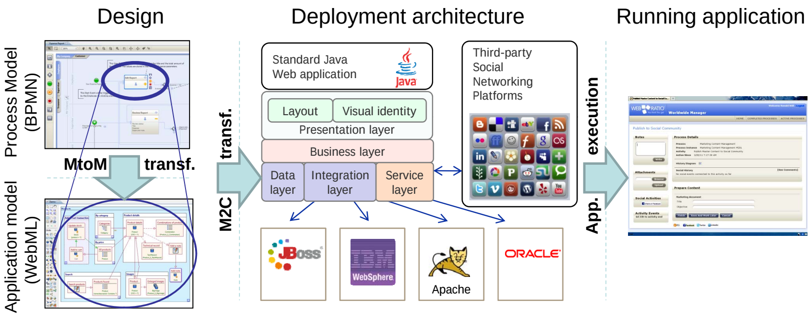


Fig. 9. Overview of the implementation of the approach within WebRatio.

7 Related Work

Social BPM goals and impact have been deeply investigated in a variety of business sectors. A prominent case is people and skill search, which has been addressed in several contexts as a means to complement data in HR systems [13]. For instance, people tagging has been applied to BluePages, the corporate directory at IBM, to improve information quality on people and skills [8]; similarly, [19] proposes an approach for building competence ontologies in a collaborative way for a repository of people's skills; [10] describes a Connection Machine for expertise finding and query answering, which updates user profiles automatically in the course of user interaction; [7] describes Fringe Contact, a system for collecting tags for an employee from his peers. Previous works on tools for social BPM focus on supporting the process modeling phase [4, 16, 21] and the execution phase [12, 17, 20, 21]. For example, [11, 16] discuss how social modeling tools allow multiple designers to register the design interactions so to produce recommendations for future projects. In the execution phase instead, the goal is to make information available to the users affected by the process even if these users were not identified at the definition stage [20]. In the process deployment field, social and business process integration is emerging also in the industry, as several vendors are proposing integrated social BPM suites. Among them, Appian, IBM BlueWorks Live, Oracle BPM Suite 11g, Software AG AlignSpace, Intalio and a few others. The approach illustrated in this paper focuses on socializing process execution, and is the first attempt at devising a linguistic instrument (Social BPMN) and a systematic method based on design patterns for expressing social process models. With respect to existing design and deployment tools, the architecture and toolsuite illustrated in Section 6 apply for the first time the pattern-driven development paradigm to the life-cycle of social BPM solutions; they support the specification of social process models and the collection and reuse of social design patterns, and provide a high level of automation to the process prototyping and deployment phases.

8 Conclusions

This paper has presented an approach for supporting the design and deployment of social BPM solutions. The core idea is to extend the BPMN visual language for process design with primitives expressing social interactions and collect the typical process socialization scenarios in the form of reusable design patterns, i.e., archetypal solutions to recurrent socialization requirements. Such social patterns give designers a systematic way of turning the socialization goals emerged during process improvement analysis into the model of a social process, which blends centralized process execution with controlled interactions with social users. The approach is backed by a BPM design tool suite, equipped with code generators and a runtime architecture, which masks the technical complexity of the integration between the process engine and the external social software. Ongoing work is focusing on the collection of a large gallery of social design patterns, on the implementation of further components in the WebRatio tool suite for integration of more social interactions and services, and on the experimentation of the methodology and of the technical approach in real-world scenarios in the enterprise and public administration, to quantify the organizational and technical benefits of a pattern-driven approach to social BPM development.

References

1. Marco Brambilla, Stefano Butti, and Piero Fraternali. Webratio BPM: A tool for designing and deploying business processes on the web. In *10th International Conference on Web Engineering (ICWE)*, pages 415–429, 2010.

2. Marco Brambilla, Piero Fraternali, and Carmen Vaca. A model-driven approach to social BPM applications. In *Social BPM Handbook*, pages 95–112. Future Strategies - WfMC, 2011.
3. S. Ceri, P. Fraternali, A. Bongio, M. Brambilla, S. Comai, and M. Matera. *Designing Data-Intensive Web Applications*. Morgan Kaufmann Publishers Inc., 2002.
4. F. Dengler, Koschmider A., A. Oberweis, and Zhang H. Social software for coordination of collaborative process activities. In *Third Workshop on Business Process Management and Social Software*, LNBIP, pages 396–407. September 2010.
5. K. Ehrlich, C.Y. Lin, and V. Griffiths-Fisher. Searching for experts in the enterprise: combining text and social network analysis. In *Proceedings of the ACM conference on Supporting group work*, pages 117–126, NY, USA, 2007.
6. S. Erol, M. Granitzer, S. Happ, S. Jantunen, B. Jennings, P. Johannesson, A. Koschmider, S. Nurcan, D. Rossi, and R. Schmidt. Combining BPM and social software: contradiction or chance? *J. Softw. Maint. Evol.*, 22:449–476, 2010.
7. S. Farrell and T Lau. Fringe contacts: People-tagging for the enterprise. In *Proc. of the Collaborative Web Tagging Workshop at WWW2006*, 2006.
8. S. Farrell, T. Lau, S. Nusser, E. Wilcox, and M. Muller. Socially augmenting employee profiles with people-tagging. In *Proceedings of the 20th annual ACM symposium on User interface software and technology*, pages 91–100, 2007.
9. Martin Fowler. *Analysis Patterns: Reusable Object Model*. Addison-Wesley, 1996.
10. M. Gker, C. Thompson, S. Arajrvi, and K. Hua. The PwC connection machine: An adaptive expertise provider. In *Advances in Case-Based Reasoning*, volume 4106 of *LNCS*, pages 549–563. Springer Berlin / Heidelberg, 2006.
11. L. Holtzblatt and M.L. Tierney. Measuring the effectiveness of social media on an innovation process. In *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems*, CHI EA '11, pages 697–712. ACM, 2011.
12. Q. Huiming, J. Sun, and H.T. Jamjoom. SCOOP: Automated social recommendation in enterprise process management. In *IEEE SCC*, volume 1, pages 101 –108, 2008.
13. E.A. Jansen. A semantic web based approach to expertise finding at KPMG. Master's thesis, Delft University of Technology, August 2010.
14. P. Johannesson, B. Andersson, and P. Wohed. Business process management with social software systems-a new paradigm for work organisation. In *Business Process Management Workshops*, LNBIP, pages 659–665. Springer Berlin Heidelberg, 2009.
15. N. Karacapilidis, E. Loukis, and S. Dimopoulos. Computer-supported G2G collaboration for public policy and decision making. *Journal of Enterprise Information Management*, 18 No 5:602–624, 2005.
16. A. Koschmider, M. Song, and H. A. Reijers. Social software for modeling business processes. In *First Workshop on BPM and Social Software*, LNBIP. 2009.
17. Gustaf Neumann and Selim Erol. From a social wiki to a social workflow system. In *Business Process Management Workshops*, LNBIP, pages 698–708. 2009.
18. K. Riemer and A. Richter. Tweet inside: Microblogging in a corporate context. In *Proceedings 23rd Bled eConference, eTrust, BLED 2010*. Paper 41.
19. A. Schmidt and S. Braun. People tagging & ontology maturing: Towards collaborative competence management. In *8th International Conference on the Design of Cooperative Systems (COOP 08), Carry-le-Rouet*, 2008.
20. R. Schmidt, F. Dengler, and A. Kieninger. Co-creation of value in IT service processes using semantic mediawiki. In *BPM Workshops*, LNBIP. 2010.
21. A. Silva, R. Meziani, R. Magalhes, D. Martinho, A. Aguiar, and N. Flores. AGILIPO: Embedding social software features into business process tools. In *Business Process Management Workshops*, LNBIP, pages 219–230. 2010.