A Model-driven Approach to Social BPM Applications

Piero Fraternali, Marco Brambilla, Carmen Vaca
Politecnico di Milano, Italy

INTRODUCTION

Social technologies have transformed the Web from a platform for passive content consumption to a place where users actively contribute to content production and opinion making. The first-generation social tools, like blogs, wikis, and content sharing applications, have been followed by a second generation of solutions, specifically focused on interpersonal interaction: social networks are intended primarily for people to meet virtually and establish networks of relations. Several variants of the concept exist, from general purpose platforms like Facebook and mySpace, to location-based social applications like Gowalla and Foursquare, to special purpose networks, like Plaxo and LinkedIn.

As the diffusion of the original Web quickly impacted professional use, pushing companies to migrate most of their B2C, B2B and B2E applications to a Web/SOA architecture and employ browser based interfaces in the work place, also the advent of the Social Web is affecting business practices. Several examples of this trend are visible: professional networking platforms have started being used by the HR departments of companies as a tool for recruiting and skill search; brand management and marketing use the viral dissemination power of social networks to test or promote products and services; Customer Relationship Management exploits enterprise blogs and social network communication (e.g., product news twittering) to install a multilateral dialogue with customers.

It is easy to foresee that the abovementioned initiatives will become systematic in enterprise management and the socialization of processes will take place both outside and outside the enterprise, affecting also those processes that today are supported by ICT tools in a traditional way. This evolution has been recently witnessed by business analysts [ORS10] who defined Social Business Process Management (Social BPM) as the effort of designing and executing business processes collaboratively.

Only a very limited number of vendors (including IBM, Oracle [Ora11], Intalio [Int11], Software AG [Sof11] and a few others) have announced strategies for incorporating social features in their BPM suites, starting from 2010. The potential Social BPM product space spans three families of solutions:

- **Traditional BPM systems**: classic BPM solutions, like Pegasystem, Savvion, Metastorm BizAg, WebRatio BPM presently do not support social interactions in business processes, but could be integrated with social tools.
- **Enterprise Social Software**: corporate Wiki, blogs and intranet portals augmented with user interaction capabilities (e.g., Intalio Social
Intranet Portal) enable social interactions, but are not meant to enact articulated business processes.

- **Emerging Social BPM suites:** early proposals have recently appeared for BPM systems integrating some social aspect.

Social BPM suites are promising but still in their infancy. For instance, IBM has announced Blueworks Live [IBM11] in mid-November 2010, but the actual social features are still under definition. Some funded research projects also work at extending traditional BPM methods towards social collaboration, such as the Prowit [Prowit11] project.

This Chapter addresses the design and development of BPM solutions that encompass communication with social tools, open-ended communities of performers, and allocation and execution of tasks to actors not known at process design time, through interactions with social network platforms. The core of the proposed approach is threefold:

- At the methodological level, providing a framework for understanding the ways to incorporate social interactions in business processes.
- At the notational level, verifying the capacity of a mainstream process modeling language (BPMN 2.0) to express social interactions and cover Social BPM requirements.
- At the technical level, exploiting model-driven software engineering techniques to produce applications enacting the social process directly from the (extended) BPMN process schema.

The main contributions of the Chapter can be summarized as follows:

- A classification of Social BPM adoption levels.
- An extension of BPMN 2.0 enabling social activities, events and conditional process flows, illustrated through a running example.
- An extension of the WebML (Web Modeling Language) for expressing Web applications interacting with social platforms, through abstract operations represented as components and design patterns.
- A technical framework for generating Social BPM applications directly from specifications encoded in the social extension of BPMN 2.0, 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 [ABBB07].

The Chapter is organized as follows: we first characterize different adoption levels for Social BPM and introduce a running example to support the subsequent illustration; then we analyze the expressive power of BPMN 2.0 for describing social features and propose some extensions, along with their syntax, semantics and examples of usage; next we illustrate a Domain specific Language for specifying Web/SOA applications implementing social BPMN features and the model-to-model transformation for mapping social BPMN specifications into application models; we briefly touch upon the implementation of the social BPMN editor and application generator in the WebRatio BPMN tool suite; and finally we draw the conclusions and highlight the future work.
SOCIAL BPM ADOPTION LEVELS

The introduction of social features in business processes can be achieved at different levels, according to a spectrum of possibilities, pictorially illustrated in Figure 1.

At the top of the spectrum, Closed BPM denotes the traditional approach supported by state-of-the-practice BPM suites. The schema of the process is decided centrally and deployed to an execution platform (e.g., a BPEL engine or a proprietary runtime). Tasks are defined rigidly, the process actors are preregistered, and allocation of actors to task follows statically defined assignment and escalation rules. The communication among the actors is channeled through the task execution interfaces, with the exception of notifications, which can be delivered through informal channels, like email and SMS.

At the next level Participatory Design opens the process design to multiple actors. Either the stakeholders can actually participate to the definition of the process model or multiple process versions are fused into one shared process model. The latter technique is relevant, for example, after merger and acquisition, when companies have to align different versions of the same process [LRDtHM11].

Participatory enactment shifts the focus of socialization from design to enactment. Although actors are fixed, as in closed BPM, the communication is not restricted to the input and output of activities but typical functions of social tools are integrated into the process enactment application to support collateral communications, like following up the status of tasks, commenting on the result of task execution, voting on quality of service, etc.

Social enactment entails opening the process execution (at least in part) to actors that are not known at process deployment time and allowing the collective execution of a task. Social task execution can take a variety of forms: from the most structured, like the use of crowd-sourcing platforms for microtasks execution, to less controlled forms, like community-based product and content rating, cooperative software development and testing, etc. The
common denominator of social task execution is the capability to launch a task to be executed by an open-ended community of performers and to monitor its progress until completion.

Finally, Process Mining is the less structured approach, where activities are executed freely and the process constraints are recovered a posteriori, by observing the behavior of the actors, e.g., inspecting execution traces [vdAPS10].

This Chapter addresses specifically participatory and social enactment.

OVERVIEW OF THE APPROACH

The goal of the described work is to provide a method for the design, fast prototyping and deployment of BPM solutions extended with social interactions, which employs high-level models for representing both the BPM requirements and the design of the process enactment application. Using models in all design stages grants important benefits:

- **Process models** are an intuitive yet formal way to represent business process, which can be shared with non-technical stakeholders and formally checked for desirable properties.

- **Application models** describe the structure and behavior of an application independently of the technical details of the deployment platform, can be formally verified for desirable properties, and used for quick prototyping and automatic code generation.

Figure 2 shows schematically the phases and ingredients of the proposed model driven Social BPM solution development.

The Process specification phase takes in input the business requirements and workflow description of an organization and outputs a model of the business processes that formally describes the organization of activities necessary for satisfying the business requirements. We assume that the process model is encoded in BPMN, extended to support the specification of social interaction. The social process model can be used directly to produce a prototype of the enactment application, whereby a business analyst or a stakeholder can: 1) impersonate any actor of the process, at all the levels of social interaction; 2) start/suspend/resume/terminate the process activities in accordance with the process constraints; 3) create and inspect project artifacts and parameters, according to the process specification; 4) impersonate external user roles and simulate social actions.

Social application design takes in input the Social BPMN model and produces in output a model of the application(s) needed for enacting it. The design phase is formally driven by model-to-model transformation rules, which dictate the way in which each (social or conventional) BPMN construct has to be turned into a construct of the social application model. The output model is encoded in the WebML Domain Specific Language, extended with social components and patterns (Social WebML). A social component is defined as a DSL primitive that represents an atomic unit of business or interface logic used to perform a social interaction (e.g., a component can support the posting of a message in a social network wall; another component can support the registration of a user to the application using her profile in a social network). A social pattern is a fragment of application design that comprises several components and connections to perform a full-fledged social activity.
(e.g., an application page for selecting an activity and posting its status on one or more social networks).

Finally, *Social process deployment* is the technical phase that produces the actual executable version of the social process enactment application. It is driven by model-to-code transformation rules that map each construct of the application model into appropriate artifacts of the deployment platform. In the case of WebML, existing rules, implemented in the WebRatio tool suite [WebRatio], map standard WebML projects onto the JEE platform. The extension to social BPM requires an enrichment of the existing model-to-code transformation rules to map the novel social WebML components into their respective implementation-level primitives. This code generation rules typically produce service calls to the Web APIs (Application Programming Interface) of the specified Social Networking Platform (e.g., the Facebook API to interact with such social network).

**Fig 2. Model driven approach to Social BPM development**

**Running Example**

We will discuss our approach through a running example inspired from a real business case described in [CFB+02] and implemented using a model-driven approach based on WebML. The business process of interest deals with the publication of contents in the B2C and B2B portals of a multinational company that sells branded PCs. The process actors comprise worldwide managers, country managers and employees of the Marketing and Communication (MarCom) department.
Figure 3 shows the BPMN representation of the process: the content production process can be initiated by the local employees that generate localized content, which can be approved by the country manager and published locally. However, the country manager may also decide that the content can be of worldwide interest. In such case, the regional-level content becomes a master copy proposal (written in English) candidate for global publication. The worldwide manager decides about the actual relevance and, based on that decision, the document is promoted to master content (and therefore assigned and translated into all the local languages) or kept at the local level (and thus approved only for local publication).

We now extend the abovementioned process by introducing (simplified) social interaction requirements, to improve the effectiveness of the process for both the internal personnel and the external customers. We restrict the example to two exemplary scenarios: B2E (Business to Employee) and a B2C (Business to Consumer).

**B2E scenario (Content Management Process)**

Let’s assume the Company has setup an Enterprise Social Network (ESN), where all employees subscribe and interact, using blogs, wikis and other social tools. The regional MarCom manager would like to take advantage of the ESN to favor the emergence of quality content in a bottom-up manner. To this end, the revised process should allow master content proposals to be commented and enriched by managers and employees of other countries, who do not see the local approval process but only the resulting content proposal. The worldwide MarCom manager follows the feedbacks and may decide to promote a “popular” content proposal to master status based on the social activities that happened upon it. As an aside to the socialization of the editorial process, the worldwide MarCom manager gets an idea of the inter-country communication paths and of the proactivity of national managers.
and employees, which can help in the selection of personnel for the role of chief editor of master content categories.

**B2C scenario (Campaign Management Process)**

In this scenario the company wishes to support the sales of collateral services (e.g., guarantee extensions and premium technical support) by letting customers enroll in the company portal through an existing social network account (e.g., Twitter, Facebook, or others) and asking them to authorize the company to post on their wall and/or to become friend. Periodically, the company launches new promotions, by posting announcements to registered customers. Customers can take active part in the campaign by proactively inviting friends to join the company’s portal, by suggesting the services, by voting the promotions, and so on. The number of successful invitations and induced sales of a user is tracked and a bonus program grants discounts when certain objectives are met. The campaign closes after a predetermined period of time and a post-campaign report is produced with performance data (new enrolled customers, online purchases in the period, etc).

**Dimensions of process socialization**

The introduction of social features in business processes affects analysis at various levels. We therefore propose a framework for classifying social process features, which can serve as a blueprint for the elicitation of social BPM requirements. Table 1 summarizes the main dimensions to consider: actor category, visibility of the process status, and level of participation.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>Values</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Actor category</strong></td>
<td>Internal performers</td>
<td>Moving from internal performers to external observers increases the intensity of social unplanned participation.</td>
</tr>
<tr>
<td></td>
<td>Internal observers</td>
<td></td>
</tr>
<tr>
<td></td>
<td>External observers</td>
<td></td>
</tr>
<tr>
<td><strong>Visibility of process status</strong></td>
<td>Direct</td>
<td>The extent to which an actor views the internal status and affects the advancement of cases and activities can be set differently for various actor categories.</td>
</tr>
<tr>
<td></td>
<td>View</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Artifact-mediated</td>
<td></td>
</tr>
<tr>
<td><strong>Level of social participation</strong></td>
<td>Inform</td>
<td>The level of participation of social actors can be modulated from pure observation to creation of signals and objects impacting on activity and case advancement.</td>
</tr>
<tr>
<td></td>
<td>Comment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Generate a signal</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Produce data</td>
<td></td>
</tr>
</tbody>
</table>

*Table 1. Dimensions to consider in social BPM analysis*

**Actor category.** The participants to a business process can be categorized based on the level of formality at which they engage in the process. Three broad categories can be recognized: 1) *internal performers* belong to the organization(s) owning the process, are formally registered before process deployment and directly affect case and activity advancement; 2) *internal observers* belong to the organization(s) owning the process, are formally registered before process deployment but cannot directly affect case and activity advancement; however, they can be notified of selected aspects of a case (e.g., events, artifacts, activity status) and may produce events and artifacts that indirectly affect case and activity advancement; 3) *external observers* do not belong to the organization(s) owning the process and may not be formally registered a priori; they can login into the process through third party identi-
fication services (e.g., through accounts at public social networks or standard open accounts); they can perform the same tasks as the internal observers and influence case advancement only indirectly. **Visibility of the process status.** The interaction with cases and activity instances can occur at different levels of control: an actor can interact via the explicit publication and manipulation of the case and activity status (*direct access*); via the right of publication and manipulation of a restricted set of case- and activity-related parameters (*view access*); or in an implicit and indirect way, via the publication and manipulation of objects or other derived information (*artifact-mediated access*). When the access to process status is indirect, the influence on activity advancement can still take place, via the mediation of a human task executed by an internal performer or a decision gateway testing some condition formally expressed in the process model. **Level of social participation.** The interaction of external social actors with a business process can produce different impacts on case advancement; external users can: be informed about the progress of a case or an activity; create comments used for (offline) process evaluation; create events that trigger the state transition of activities; create data objects or modify parameter values that alter the status of activities or the control flow of the case.

Table 2 summarizes the social BPM dimensions relevant to the analysis of the running example.

<table>
<thead>
<tr>
<th>Actor Categories</th>
<th>Worldwide manager, Country managers and Local employees involved in a Content Management Process case</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal performers</td>
<td>Worldwide manager, Country managers and Local employees involved in a Content Management Process case</td>
</tr>
<tr>
<td>Internal observers</td>
<td>Managers and employees of other countries, not assigned to or owning the active Content Management Process case</td>
</tr>
<tr>
<td>External observers</td>
<td>Customers self-registering in the portal for the Campaign Management Process</td>
</tr>
<tr>
<td>Visibility of Process Status</td>
<td>Worldwide manager, country managers and employees acting on activities of the Content Management Process case they own/ participate in</td>
</tr>
<tr>
<td>Direct access</td>
<td>Worldwide manager, country managers and employees acting on activities of the Content Management Process case they own/ participate in</td>
</tr>
<tr>
<td>View access</td>
<td>Managers and employees of other countries viewing and commenting content proposals produced in a non-owned process case</td>
</tr>
<tr>
<td>Artifact-mediated</td>
<td>Customers receiving campaign announcements (read access) and creating user invitations (write access)</td>
</tr>
<tr>
<td>Level of Social Participation</td>
<td>Worldwide managers receiving notifications of comments to content proposals</td>
</tr>
<tr>
<td>Inform</td>
<td>Worldwide managers receiving notifications of comments to content proposals</td>
</tr>
<tr>
<td>Comment</td>
<td>Country managers and employees reading and commenting content proposals</td>
</tr>
<tr>
<td>Generate a signal</td>
<td>Worldwide managers promoting content proposals to master content status</td>
</tr>
<tr>
<td>Produce data</td>
<td>Portal users creating friend invitations</td>
</tr>
</tbody>
</table>

*Table 2. Social dimensions in the running example*
SOCIAL REQUIREMENTS FOR BPMN

BPMN is one of the most complete visual languages for expressing business process models. The current version (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. Tables 3, 4, and 5 illustrate how BPMN can be used or extended to represent the essential aspects of socially enacted processes. In particular, Table 3 presents the requirements associated to the implementation of social actions at large (at process and flow level); Table 4 describes the requirements at the single task level; and Table 5 describes the requirements at the event management level.

The main extension is the concept of Community Pool, which is defined as the pool devoted to social activities: it may represent a public social networks or an enterprise social network.

<table>
<thead>
<tr>
<th>Requirement</th>
<th>BPMN notation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actor category and community pool</td>
<td><img src="image" alt="Icons" /></td>
<td>Annotation in the pool/lane header, respectively for: Internal performer, internal observer, external observer. The last two categories characterize community pools</td>
</tr>
<tr>
<td>Activity inheritance</td>
<td>Hierarchical definition of user roles</td>
<td>To avoid duplication of activities, hierarchies of roles with inheritance of task execution can be defined</td>
</tr>
<tr>
<td>Community executed activity</td>
<td><img src="image" alt="Ad hoc Activity within a community pool" /></td>
<td>A sub-process or a task delegated to a community of observer users. Ad hoc mark denotes complex / non-structured interactions</td>
</tr>
<tr>
<td>User vs. Automatic Gateways</td>
<td><img src="image" alt="User vs. Automatic Gateways" /></td>
<td>The execution of the choice is performed by a human or by an automatic procedure /rule</td>
</tr>
</tbody>
</table>

Table 3. Social requirements for BPMN at pool and process level.

Table 3 describes social BPMN extensions at pool and process level:
- **Actor categories** are denoted by proper icons in the pool/lane header;
- The **hierarchical definition of user roles** is provided to avoid duplication of activities across lanes. This is a general requirement, but gets particularly critical in social applications where multiple roles may perform social tasks; to avoid specifying activities repeatedly in every lane, a parent role factoring out the activities common to multiple sub-roles can be used, which makes the specification of social behavior more compact.
- The BPMN 2.0 concept of **ad hoc parallel** task is proposed as an effective solution for describing social activities, which are inherently unstructured;
• **User gateways** are introduced to express human choices. The specification of human-executed choices in BPMN is an open issue currently under discussion in the BP analyst community. Social applications corroborate the need of a better way to explicitly model human decisions, which can be achieved simply by distinguishing *user gateways*, which entail a user's decision; and *automatic/service gateways*, which entail a decision taken according to a rule (this is the standard semantics for BPMN gateways) or performed by a service.

Notice that the visibility of process status and the level of social participation do not require an ad hoc notation because they can be inferred from the BPMN diagram, by looking at the flow, task, and event definitions. While most social interaction may be described with standard message flows within the community pool or between the community and the enterprise pool, a simplified notation may be convenient in several cases. Table 4 proposes a set of annotations describing social interactions at the task level. These annotations are a shortcut for social activities: they express 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). These annotations greatly simplify the expression of typical social activities, because they avoid the visual clutter due to the introduction in the BPMN diagram of the community pool and to the plethora of messages flowing into/from it.

Finally, Table 5 shows the event types needed for supporting 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.

---

<table>
<thead>
<tr>
<th>Requirement</th>
<th>BPMN notation</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task Type: Social broadcast</td>
<td>Annotation of task</td>
<td>Data flow to a community pool</td>
</tr>
<tr>
<td>Task Type: Social posting</td>
<td>Annotation of task</td>
<td>Data flow to a single user in a community pool</td>
</tr>
<tr>
<td>Task Type: Invitation to activity</td>
<td>Annotation of task</td>
<td>Dynamic enrollment to a task in the process case</td>
</tr>
<tr>
<td>Task Type: Commenting</td>
<td>Annotation of task</td>
<td>Submission of comments about the activity</td>
</tr>
<tr>
<td>Task Type: Voting</td>
<td>Annotation of task</td>
<td>Voting (y/n) on an activity, either within a social network platform or directly in the BPM system</td>
</tr>
<tr>
<td>Task Type: Login</td>
<td>Annotation of task</td>
<td>Use a social profile to login into a system</td>
</tr>
<tr>
<td>Task Type: Invitation to join a network in a community (e.g., friendship)</td>
<td>Annotation of task</td>
<td>Invitation between community members to join a specific network (e.g., friend network, group, and so on)</td>
</tr>
<tr>
<td>Task Type: Search for actor’s information</td>
<td>Annotation of task</td>
<td>Lookup query to the community to search for actor with specific profile attributes</td>
</tr>
</tbody>
</table>

*Table 4. Social requirements for BPMN at task level.*
Table 5. Social requirements for BPMN at event level.

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

**Social BPMN specification of the running example**

To exemplify social BPMN extensions, Figure 4 shows the BPM model of the social B2E scenario of the running example: with respect to the original model, displayed in Figure 3, the process is extended with the publication of the Master content proposal to the social community. Notice that the activity is annotated with the **social commenting** stereotype (as from Table 3), to specify that a request for comments is issued to the social network. The activity accepts non-blocking social events (representing the received comments). The activity runs for a week (as marked by the temporal event). Furthermore, if the Master content is actually promoted, the new state is notified the community through a **social broadcasting** stereotyped activity.

**Fig. 4. Social BPMN extensions applied to the B2E scenario (Content Management Process).**
To exemplify the B2C scenario, we describe two important processes of the Campaign Management Process presented in the running example. Figure 5 shows the user registration process, representing the log on phase exploiting a social user profile. The user registers in the company portal using his credentials in a social network; the company gets the enrollment social event and automatically replies with a request for friendship (or authorization for any other kind of social interaction) posted in the user’s social network. If the user accepts, the company updates his profile in its member directory (upon reception of a friendship confirmation event) and will exploit this information for future promotional campaigns.

![Fig. 5. User enrollment and authorization in the B2C example.](image)

Figure 6 shows the Social Campaign Management process, which implies an explicit interaction between the company’s MarCom personnel and the observer users clustered in a social pool. The process initiates when the company’s marketing manager decides that a new campaign has to be launched; a marketing employee prepares the campaign content and broadcasts the event (together with the content) to the social pool with an explicit message flow; as a consequence, each user that previously accepted to receive campaign announcements will get a message in her favorite social network applications. The process then continues within the social pool. A social network member may comment back to the company, suggest the service/product to his own friends, or invite new friends to the company network or social group. As a result, new actors can enroll to the company’s portal and dynamically become part of the social pool. Each action from a social pool actor is notified to the company, which in turn registers the events for the whole duration of the campaign. If the user reaches the agreed objectives, a task is activated that notifies the reward from the company to the user in the community pool, by means of an individual communication in the user’s preferred social network(s). The campaign terminates at the end of a designated time interval; after the temporal event, the marketing manager can analyze the final report on the campaign results.
Notice that an explicit BPMN representation of the social task is used in this case, to describe in detail what happens in the social community pool; this requires using a parallel ad hoc task inside the community pool. Message flows are also shown explicitly to describe the message sources and contents. On the other hand, the “send rewards to user” activity is specified using the task stereotype notation, because the task behavior is standard and the message follows can be deduced from the short notation and from rest of the process diagram.

Fig. 6. Social Campaign Management in the B2C social example.
DESIGN OF SOCIAL APPLICATIONS WITH WEBML

As shown in Figure 2, once the BPMN specification is completed, application models are designed that describe the software solutions required for enacting the process; such application models specify, for each role involved in the social process, the interface, interaction commands, service calls, and data exchange flow necessary to carry out the tasks specified for that role in the process. When the Web is elected as the platform of choice for the implementation of the process enactment solution, the application model can be naturally encoded in the WebML notation [CFB+02], a visual language for designing data- and service-intensive Web/SOA applications. A WebML model consists of one or more site views, which represent hypertext application used to publish or manipulate data and interact with the back end business logic. A different site views can be defined for each process actor; internally, a site view consists of a set of pages, atomic units of interface, containing units, representing data publishing components. Units are related to each other through links, representing navigational paths and parameter passing rules. Additionally, the WebML application model comprises the definition of Web services (when needed) and their invocations.

Starting from a BPMN specification, an automatic model-to-model transformation produces the WebML model of the process enactment applications of each role, and a database schema for storing the case advancement data. The standard model-to-model transformation from BPMN to WebML is already supported in the WebRatio BPM commercial tool suite [ABB07]; this has been augmented to cover the social extensions of BPMN described in this Chapter, thanks to three new features: 1) the social BPMN primitives of Table 3, 4 and 5 are added to the BPMN editor; 2) novel components modeling the application design patterns and software components for interaction with social networks are added to the WebML languages, using its native extension mechanism; 3) the BPMN-to-WebML mapping rules are enriched to map each social BPMN primitive into the appropriate social WebML design pattern.

Example of social BPMN to WebML mapping and quick prototyping

To exemplify how the social BPMN model is mapped to WebML, we show a simple activity with social effects, the WebML model produced from it, and the quick prototype generated by WebRatio.

Figure 7(a) shows the BPMN model of a task publishing some materials and waiting for comments from a social network (for a period up to one week). Figure 7(b) shows the WebML application model generated for the task: the entry point to the task is expressed by the Input unit, which is the component that collects the input parameters of the task from the case enactment database. The Publish to the social community page denotes the home page of the task, which retrieves the task parameters from the input unit and displays a form (Prepare content unit) allowing the user to enter the content to be submitted to the social network. Once the content is ready, an outgoing link from the Prepare Content unit permits the user to trigger a sequence of operations: the social network API is first invoked to publish the content; next the task parameters are updated and finally the task is suspended. The user can come back at any time and resume the task for looking at the comments received so far (these are displayed in the Comments index unit.
within the *See Comments* page). The activity terminates after 1 week as requested in the BPMN model.

![Diagram](image)

**Fig. 7.** (a) BPMN notation for a social-based activity; (b) Corresponding WebML application model; (c) Visual rendering of an activity UI in the prototype generated by WebRatio.
Figure 7(c) shows the snapshot of the Publish to social community Web page generated as a prototype from the WebML model of Figure 7(b). The user can see: his name (Roland Still), his role (Worldwide Manager), the details about the process and the task together with the correlated social history registered so far (in the Process Details box). A link leads to the See Comments page. The actual inputs required by the current process step must be submitted in the Prepare Content and Attachment boxes. The Social Activity box contains the possible actions that can be performed at the community level. In the example, the user can share the outcomes on Facebook (the social network of choice). The Activity Events box displays the list of BPM events to be handled (in the example, the deadline for closing the activity). Finally, in the bottom part of the page, the list of all the social platforms involved in the whole application is listed (and the one of the current task is highlighted). By clicking on the various icons, the user can see the respective events happened so far.

**Implementation within the WebRatio tool suite**

The described generative framework has been implemented as an extension of the WebRatio Model-Driven Web application environment, which enables developers to edit WebML models and automatically transform them into running applications for the JEE and Service Oriented Architectures. WebRatio is commercially available and has been used to produce large scale Web and BPM solutions; the BPM edition, presented at the BPM 2010 Conference (Hoboken, NJ) [BBF10], comprises: a BPMN editor, a code generator that includes the transformation from BPMN to WebML and a quick prototyping function that produces JEE-compliant process enactment applications (one-click prototyping). This functionality invokes in sequence the two transformations from BPMN to WebML and from WebML to JEE, yielding a dynamic, multi-actor enactment application with a default look & feel. The generator creates a few exemplary users for each BPMN actor, which allows the analyst to impersonate each role in the process. We are now extending this framework to support the newly introduced social features: BPMN social artifacts, new WebML components implementing the different behaviors and the interaction with various social networks APIs, and the new model transformation rules that generate the WebML models from the social BPMN notation. As a result, the prototyping function will deliver in one-click social process enactment applications, whereby the analyst can actuate both the structured and the social part of the BPMN specification, impersonating all roles: internal performers, internal observers and also external users in a social community of choice. The rapid prototype validated in this way can then be evolved by the WebML designer into the final version of the process enactment applications, deployable on top of any industrial-strength JEE platform.

**Conclusions**

This chapter has discussed the levels of adoptions of social features inside organizations and business process models, and examined the requirements that Social BPM poses to process specification languages. Based on that, we have illustrated a set of extensions and exemplary scenarios for designing social BPM applications with BPMN 2.0 and briefly described the technical architecture of a tool suite for the generation of rapid prototypes directly from socially-extended BPMN models. Our future work will concentrate on
completing the implementation of the social extension of WebRatio BPM, validating it on real-world industrial cases, and quantitatively measuring the benefits that process socialization brings to organizations, by means of novel Business Analysis Monitoring dashboard applications that take into account both the structured and the social part of a business process.

REFERENCES


EXAMPLE OF INDEX
Actor Category, 8
Application model, 4
BPMN, 2, 4, 6, 9, 10, 11, 12, 13, 14, 15
Business Process Management System (BPMS), 1
Enterprise Social Software, 1
IBM Blueworks Live, 2
Process Mining, 3
Process model, 4

reputation, 12
Social BPM, 2
social participation, 4, 8, 10
Social process deployment, 5
social relationship, 12
social tool, 1
Web Modeling Language (WebML), 2, 4, 5, 6, 14, 15
WebRatio, 14