

Workflow Enactment in a Social Software Environment

Davide Rossi and Fabio Vitali

Dipartimento di Scienze dell'Informazione
Università di Bologna, Italy
{rossi,fabio}@cs.unibo.it

Abstract. Originally conceived with different goals in mind, both Business Process Management (BPM) tools and social software applications help organizations in enacting multi-actor processes. The paradigms they are inspired to are, however, very different and this is mostly true if we focus on how coordination among actors is carried on. In a BPM context, usually, the process that has to be enacted is well defined: a model of the process exists and the interactions among actors are enforced by this model. We refer to this approach as *structured coordination*. Social software, on the other side, fosters the enactment of processes by putting collaboration tools into the hands of the users. In this case we witness to a significant example of what we call *emerging coordination*. While a synthesis of the two paradigms is arguably impossible, it is however interesting to study how BPM and social software can benefit from each other. In this paper we show how principles of structured coordination can be injected in a social software environment in order to enact simple workflows. To this end we introduce Social X-Folders, and extension of X-Folders designed to interact with social software tools.

1 Introduction

In the last few years an increasing number of companies are adopting social software tools to support their business processes, a trend called *Enterprise 2.0* [7]. BPM tools have been the only viable solution in this context for a long time, but now they have to face a strong competition. Social software tools are easier to set up and maintain and do not require any modeling effort, they allow to address dynamic changes and reconfigurations, exception handling, flexibility and, most importantly, knowledge workers can reclaim full control on the business processes. Human factors and cultural resistance are well known limiting factors for BPM tools [3] that have a much smaller impact on a social software environment. Of course there is a price to pay in terms of guarantee of progress, monitorability, traceability, that makes the social software solutions reasonable only for non-critical processes. But in many cases the advantages of the social software approach are enough to push managers to demote processes (or, at least, sub-processes) to the non-critical rank. Not all limitations, however, can be worked around by relaxing the constraints of the stakeholders, and this is the

case for the lack of mechanisms to automate part of the activities (even in most human-centered workflows same tasks can be carried out automatically). While some of the existing social software tools can enact small workflows including some kind of automatization, this applies only within the tool itself. One of the main strengths of the social software approach, however, is that of providing users with a wide array of different tools (blogs, wikis, forums, shared calendars and so on) that can be used synergically, and no existing tool or service allows the enactment of workflows spanning across multiple applications. The reason behind this limitation is mainly related to a paradigm shift: the adoption of social software also entails a switch from structured to emerging coordination. Structured coordination is process-first: the process is well-defined; usually a model that describes it exists and interactions among actors are predefined. Emerging coordination is tools-first: the actors have a set of tools and they use them to interact; the workflow is not well-defined (if defined at all) so IT support becomes difficult. But having no predefined process model does not mean the chaos, what usually happens is that, after the first few iterations, the process “takes shape”, becoming more and more structured. In this paper we introduce Social X-Folders, an extension of X-Folders [8] designed to model and enact workflows in a social software environment. X-Folders is a rule-based workflow system that can be used to enact multi-actors, distributed workflows, by using a peer-to-peer approach, a paradigm that fits well in the social software context. Social X-Folders can also assist in the gradual automatization of processes during their refinement, supporting the natural evolution of coordination patterns typical of social software.

2 X-Folders

X-Folders is a software environment for multi-party document-based processes that aims at supporting the implementation of workflows involving multiple users that interact by means of documents stored in special, reactive, folders: the X-Folders. A reaction in X-Folders is a program whose control flow depends on the status of the folder and whose actions are invocations of local services (that can be used to interact with the documents in the folders) or of services exposed by external components. The actions performed by the actors and the reactions fired by the system flow in a sequence that drives to the accomplishment of the workflow process coordinating humans and software components using documents. X-Folders is not meant for implementing mission-critical processes, but rather to ease the implementation of light workflow processes between peers, with the advantages of a real distributed architecture and a relatively small run time system.

3 X-Folders in a Social Software Environment

X-Folders has been designed for a context in which all the knowledge the user has access to is contained in documents, intended as bodies of information that

can be physically encoded into files. This is not what usually happens to a user in a social software context. Pieces and bits of knowledge can be spread among different sources: forums, blogs, wikis, podcasts; the focus shifts from the space of the documents to the *space of knowledge*. The problem of dealing with multiple information sources is well known in the social software community and is addressed mainly by using feeds and aggregators. By using feed technologies like RSS and Atom, web syndication can help in monitoring different information sources; web aggregators help building a personal information space out of several feed sources. In this environment it makes sense for X-Folders to be based not only on reactive document folders but also on reactive knowledge spaces, populated using syndication techniques. Mapping this concepts to X-Folders is quite straightforward: a knowledge space can easily be represented by using a smart (or virtual) folder in which feeds are sub-folders and entries (or items) in it are documents. This solution allows X-Folders to address the specificities of the social software ecosystem by using its own tools and technologies. Just like reactive document folders monitor the state of the contained documents and reacts when they reach a certain state, smart X-Folders monitor different information sources. Feed aggregation thus address the “firing part” of X-Folders, i.e. the rule activation logic. What is still missing is the “active part”: what kind of actions can be fired? We extended the local X-Folders web services with simple ones that allow to perform usual social software-related activities, like writing an entry in a blog or posting a message in a forum. This solution fits nicely into the X-Folders framework providing an extension that maintains the advantages of the original system and allows its integration in a social software environment. With respect to the social software tools, in fact, the workflow engine is just an actor operating on the behalf of the user. This means that no specific awareness is required from the tools, limiting the intrusiveness of the solution. The resulting architecture can be seen as an instance of PageSpace [4], a reference architecture for multiuser, interactive applications that we conceived more than ten years ago.

3.1 Architecture and Technological Solutions

In order to implement the features outlined in the previous section, the X-Folders architecture has been enriched with a feed aggregator that extends the storage manager in order to support smart X-Folders. In the existing prototype implementation, the access to smart X-Folders can only be performed by using the SOAP interface but it is possible to extend the WebDAV interface in order to allow read only access. Since web syndication technologies are pull-based (meaning that updates have to be fetched from the client and no server-notification function is available), polling techniques are used to update the feeds periodically. Reaction rules associated to a smart X-Folder are evaluated after each update. Since subsequent activations of already fired rules for the same entry in the feed turned out to be a recurring problem when defining the rules, the standard behavior of X-Folders has been slightly modified when checking the

firing of rules associated to smart X-Folders: the rule is activated only when the firing predicate toggles from false to true.

New web services are also made available by the local web service provider in order to interact with social software tools. Since most existing social software tools do not provide a programmer’s API, most of the functions supported by the new web services are implemented by connecting to the web site of the tool and using HTTP “pretending” to be a human user interacting with a regular web browser. This solution is largely sub-optimal: changes in the sequence of web pages dispatched by the tool, or even simple changes in the names of the forms elements, easily lead to a malfunctioning of the system. This has long been a problem with the web: gathering information has been much easier than publishing it. Some of the social software tools (like the wikis) have been developed to overcome this limitation but, ironically, the very same problem arise now with wikis in a semantic web context. We can only hope that in the future a standardized approach to programmatically interact with social software tools will emerge, complementing existing syndication techniques; we are quite confident this is going to happen quite soon because of the need of the enterprises adopting these tools to integrate them with their existing ones.

The high-level architecture of a Social X-Folders site is depicted in fig.1.

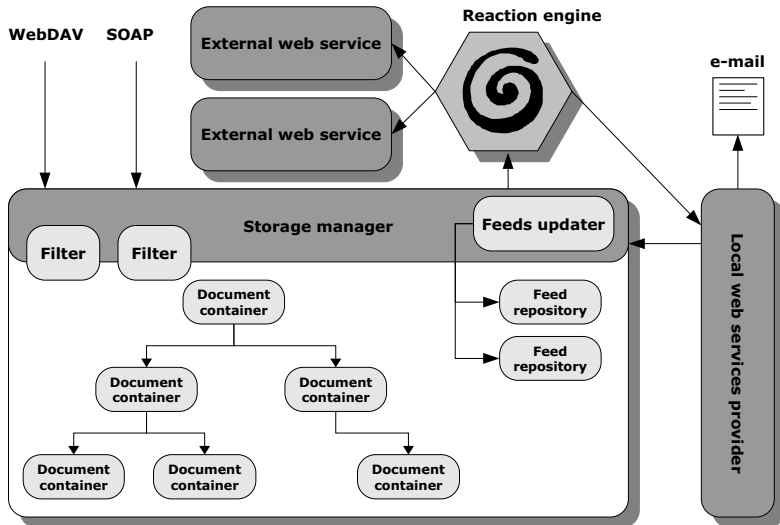


Fig. 1. The high-level architecture of an X-Folders site

4 Proof of Concept

As a very simple proof-of-concept we implemented a “personal workflow” by using Social X-Folders. The idea is to automate as much as possible the tasks required to manage a photographic contest using a forum. When a new contest

starts, a new thread is created in the contests forum. Participants can submit their photos by adding posts in this thread. The thread remains accessible for a specified amount of time (e.g. a week), after that the thread is locked and a new poll thread is created. This latter thread is used to collect the votes from other participants. After another specified amount of time, the poll thread is also locked and the poster that received most votes is the winner. A workflow like the one we just outlined can be implemented in very different ways by using Social X-Folders, the solution we propose here has the merit of showing how several social software tools can be orchestrated. In this solution we use a shared calendar tool (Google calendar) to set up the events that let the workflow progress (like begin new contest, lock submission thread, create poll thread and so on). A feed associated to these calendar events is represented by a specific smart X-Folder. Reaction rules associated to this folder fire when the events in the calendar are scheduled and, by using the local web services, they interact with the forum application in order to implement the required actions.

As an example, the reaction rule used to start a new contest thread is as follows (“...” are used to replace example-specific parameters that have been removed for clarity):

```
<when trigger='/feed/entry[title
    [starts-with(text(), "New photo contest:")]
    [content[ends-with(text, "Status: confirmed")]]'>
  <call name="phpBBCreateThread">
    <param name="forumURL">...</param>
    <param name="forumId">...</param>
    <param name="threadTitle">xpath($this/title/text())</param>
    <param name="message">../protoFolder/contestMessage</param>
  </call>
</when>
```

In the existing prototype these rules have to be written by hand (and require a skilled user to be set up correctly); in the future wizard-based web interfaces could be used to ease this task.

The very simple workflow example introduced in this section is meant to show the main characteristics of the system but we would like to point out that by using several X-Folders systems it is possible to set up more complex distributed multi-actor workflows by using a peer-to-peer approach.

5 Discussion and Related Works

X-Folders has been designed in order to provide end-users with a simple yet powerful tool that, using standard web-related technologies, can assist them in enacting simple workflows that can scale using a peer-to-peer approach. Social X-Folders is a reasonable extension that introduces concepts related to structured coordination in the social software experience. In its design we followed

an *exogenous* approach, meaning that we operated by layering an automatic enactment service on top of existing social software tools and not by modifying them. We choose this solution for several reasons. First of all injecting *process-awareness* into the code of several social software tools can turn out to be a very complex task. Moreover in a Software as A Service (SaaS) context [2], which is often the case with Web 2.0, the option of modifying the software tools is simply not available. The obvious price to pay is that we cannot enforce actors not to perform specific interactions with the tools (such as moving a document, deleting a post, ...) also when the current state of the process would suggest otherwise, but this falls outside the scope of the current papers which is about supporting automatic enactment.

The value of this experiment with Social X-Folders is twofold. First, it shows that elements of structured coordination can bring interesting advantages to an emerging coordination environment; unsurprisingly this has been accomplished by using a rule-based coordination approach [6] (which is closer to the principles of social software with respect to the approach based on a (graphical) modeling language). Second, it shows that it is possible to create artifact-based workflows in a context in which artifacts are complex entities obtained by aggregating multiple sources. This last consideration finds its relevance in the idea that processes mixing BPM tools and social software tools are probably bound to find interaction points around artifacts, and artifacts in a social software environment are often bits and pieces of scattered information. For example a large business process, enacted by BPM tools, could include sub-processes meant for being enacted with social software tools. A problem that arise in this context is to determine when a “social” sub-processes has terminated its execution. Given the fact that no process status is directly modeled in a social software environment, this kind of decision has to be taken by analyzing the artifacts of the process and aggregators can turn out to be very useful to this end.

The merging of BPM tools with tools designed to help users in collaborative processes has already been addressed in the past when the CSCW research community tried to find a synthesis of workflows and groupware software [1] [5]. The context, however, is quite different. While social software and CSCW/groupware tools share common traits in that they both address the interaction and the collaboration among users, they are not based on the same paradigms. CSCW embraces the top-down approach to coordination (by predefining the collaboration rules among actors) while social software promotes the bottom-up one.

The idea of introducing workflow support in social software is not new and most content management tools (like the most complex wikis and blogs) implement some kind of document workflow. What make Social X-Folders apart from the rest is that it is not meant to support a workflow within a specific tool but, rather, to enable workflows spanning across different tools. In the last few months we are witnessing an increasing number of applications and services meant to assist people in their social software *experience*: passive aggregators are developing into personal information management systems, the introduction of processes and workflows seems a natural step in this direction and we should

not be surprised when major aggregators, like Google, will begin to provide this kind of service in the future.

6 Conclusions and Future Work

BPM tools and social software tools are based on very different paradigms but share common goals. While a synthesis between structured and emerging coordination is arguably impossible, BPM tools and social software can complement each other. It is also possible to introduce concepts related to BPM into a social software environment and vice versa. In this paper we have shown an experiment in this direction. In the future we plan to refine Social X-Folders in order to provide social software users with a service to manage their processes space just like web aggregators manage their information space. This means moving X-Folders into a full web application with no software to be deployed into users' machines. This solution would also better comply with SaaS philosophy. We are also interested in analyzing case studies in which BPM and social software tools can be deployed side by side in the context of the same overall process with specific interest in (artifact-based) synchronization mechanisms among the two environments.

References

1. Ben-Shaul, I., Kaiser, G.: Integrating groupware activities into workflow management systems. In: Proceedings of the Seventh Israeli Conference on Computer Systems and Software Engineering, pp. 140–149 (June 1996)
2. Bennett, K., Layzell, P., Budgen, D., Brereton, P., Macaulay, L., Munro, M.: Service-based software: the future for flexible software. In: APSEC 2000: Proceedings of the Seventh Asia-Pacific Software Engineering Conference, Washington, DC, USA, p. 214. IEEE Computer Society, Los Alamitos (2000)
3. Bernstein, A.: How can cooperative work tools support dynamic group process? bridging the specificity frontier. In: CSCW 2000: Proceedings of the 2000 ACM conference on Computer supported cooperative work, pp. 279–288. ACM, New York (2000)
4. Ciancarini, P., Tolksdorf, R., Vitali, F., Rossi, D., Knoche, A.: Coordinating multi-agent applications on the www: A reference architecture. *IEEE Trans. Softw. Eng.* 24(5), 362–375 (1998)
5. Grundy, J.C., Hosking, J.G.: Serendipity: Integrated environment support for process modelling, enactment and work coordination. *Automated Software Engineering* 5(1), 27–60 (1998)
6. Kappel, G., Rausch-Schott, S., Retschitzegger, W.: Coordination in workflow management systems - a rule-based approach. In: *Coordination Technology for Collaborative Applications - Organizations, Processes, and Agents [ASIAN 1996 Workshop]*, London, UK, pp. 99–120. Springer, Heidelberg (1998)
7. McAfee, A.P.: Enterprise 2.0: The dawn of emergent collaboration. *MIT Sloan Management Review* 47(3), 21–28 (2006)
8. Rossi, D.: X-folders: documents on the move: Research articles. *Concurr. Comput.: Pract. Exper.* 18(4), 409–425 (2006)