Lightweight Process Modeling for Virtual Enterprise Process Collaboration

Lai Xu[1], Paul de Vrieze[1], Keith Phalp[1],
Sheridan Jeary[1], and Peng Liang[2]

[1] Software Systems Research Centre, Bournemouth University, UK
{lxu,pdvrieze,kphalp,sjeary}@bournemouth.ac.uk
[2] State Key Lab. of Software Engineering, Wuhan University, China
liangp@sklse.org

Abstract. Mashup is a new web 2.0 technology for data aggregation applications, combining data from different sources to create valuable information. The uses of mashups are often more data related than process related. In this paper, we explore the differences between data-oriented mashups and process-oriented enterprise mashups and consider how process mashups can be used for virtual enterprise collaboration. We highlight the modeling of end users' process mashup applications from both a control flow and a data flow perspective. Based on our analyses, a lightweight process modeling approach is proposed for process enterprise mashup applications. Our approach, illustrated by reference to an example personal collaborative activity, will support collaboration among users with different levels of modeling skills and expertise in a virtual enterprise environment.

Keywords: Lightweight business process modeling, Business process modeling, Virtual enterprise process collaboration, Process-oriented mashups, Web 2.0.

1 Introduction

The internet lies at the core of a connected world, acting as a conduit for the exchange of information, allowing tasks to be processed collaboratively, and enabling the formation of communities amongst users with similar interests. An Internet interconnected world has increased both business and personal efficiency and performance. On-line mashups have the capability to combine data with (possibly external) functionality to create and produce useful outputs. The term “mashup” implies easy, fast integration, usually made possible by access to open APIs and data sources to produce results that the data owners did not originally envisage.

As the business environment changes rapidly the ability set up a collaborative business process in a virtual enterprise is desirable. Collaborative business processes are increasingly driven by business agility, adaptability, and flexibility, particularly in a virtual enterprise environment. There is increased pressure to build enterprise applications quickly in order to respond to situational needs of the business.

Service-orientation allows a way of thinking of business process management in terms of computational infrastructures, services, service-based development and outcomes

L.M. Camarinha-Matos et al. (Eds.): PRO-VE 2010, IFIP AICT 336, pp. 501–508, 2010.

of those services [1]. Service-oriented architecture (SOA) is a significant computing paradigm and is being embraced by organizations worldwide as the key to business agility. Web 2.0 technologies such as AJAX enable efficient user interactions for successful service discovery, selection, adaptation, invocation and service construction. SOA and Web 2.0 technologies also balance automatic integration of services and human interactions, separating content from presentation in the delivery of the service. Another Web technology, such as Web services, implements functionality using pre-designed building blocks. Integrating SOA, Web 2.0 technologies and Web services into a service-oriented application connects business processes in a horizontal fashion.

We propose a process-oriented enterprise mashup [2], [3], and [4], which allows a user to specify their needs, find related web resources, and eventually execute the resulting process for rapidly building collaborative business process in a virtual enterprise. The first step is to allow business users who have no professional modeling skills to model business processes. Thus, lightweight process modeling for process-oriented mashups aims to be a step towards the enablement of business process modeling to an end user group in a virtual enterprise. The finished business process models provide the basis for process automation.

1.1 Related Work

Well established business process modeling languages like Petri net, Event Process Chains (EPC), Business Process Modeling Notation (BPMN), or Business Process Execution Language (BPEL) require major training efforts and thus are only accepted by certain groups of people. Our work aims to enable the end-user and introduces an intuitive and guided way to model business processes. One facet which this work addresses is the context awareness aspect of business process models to achieve reusability and consistency. Due to page limitation, we are not going details such as how to implement context-awareness modeling principle.

Papers [5] and [6] propose a new modeling approach based on proclets, performatives and channels. A proclet represents only one aspect or one element of a whole business process using Petri net style notations. Our lightweight process language can also be used to describe a proclets or used as a requirements acquisition tool to describe a part of a whole process.

Our previous work [2], [3], and [4], has identified key issues in the implementation and use of a lightweight business process modeling environment. One of the key issues is in supporting end-user process modeling. Papers [7] and [8] discuss lightweight business process modeling issues only from a control flow perspective. In this paper, we examine the capabilities that need to be added to data-oriented mashups to enable the implementation of process-oriented enterprise mashups. We first identify the modeling requirements of lightweight process applications in Section 2 and introduce the modeling principles for process-oriented mashup applications in Section 3. This is followed by discussion of details of process modeling related with both a control flow and a data-flow perspectives in Section 4 and Section 5 respectively. A personal collaborative activity is used to demonstrate our modeling method in Section 6.

2 Lightweight Business Process Modeling Requirements

Lightweight process modeling is a combination of techniques that seek to lower the barrier to entry for users who need to apply some process modeling to their current task. The approach involves fostering a participative style of modeling and providing a forum for community of experts. However to enable a user to participate in lightweight process modeling and execute the process models there are a number of technical requirements.

The fist requirement is to provide executable process models. A process-oriented enterprise mashup should allow a user to specify and automate business tasks and activities. Hence, the lightweight business process modeling language must be executable. The modeling language needs to provide logic information which is associated with precise semantics that can be used to automatically validate and simulate business processes.

The second requirement is providing easy access to the modeling environment for end users and to ensure business users buy-in there needs to be a focus on usability. This comprises both simplicity of the approach and a low foot-print for the solution ("zero install"). Ideally, users can model processes using simplified notations in a Web 2.0 environment and draw up on modeling best practices.

Actively participation is the third requirement for supporting lightweight process modeling. Combine both structured and unstructured information and publish them as a single point of reference to the organization, such as blogs or wiki. A process wiki allows the community to participate in discussions, and to provide comments or ratings in relation to process proposals. Process documents can be generated for offline reading and dissemination.

The process editor should provide an intuitive user interface. Furthermore, it needs to ensure the user is given advanced guidance during the modeling activities. Errors, misspelling and inconsistencies should be avoided from the beginning.

3 Lightweight Business Process Modeling Principles

Following the summary of the requirements for lightweight business process modeling, it is possible to detail our four modeling principles for process-oriented enterprise mashups. The first principle is to minimize the need to design from scratch. For many end users, it is difficult to start to create a model from scratch and it is easier to design a process model based on an existing example. Learning by examples style is common in a community environment where pre-design, highly reusable process models and templates can be selected. These existing models can also serve as a learning model or a base for end users to create a new process model.

The second principle relates to context awareness. The names of activities and tasks involved in a process model should be unified. The context-driver principle allows identification, storage, and representation of a business process artifact only once. A business process is instantiated depending on specific context categories (e.g., business process, industry, country, business role, etc.).

The third principle involves the concept of reusability. We adopt eight control flow patterns from workflow patterns. Details can be found in Section 4. Process patterns, process templates, process fragments, reference models, best-practice models, and further example models are provided in association with the process modeling language; these are all reusable.

Finally, the fourth principle is the community. A web-based community environment provides recommendations, ranking, accompanied by commenting on process models and templates. These process templates will initially be related to usage in the early stages of development, but we expect them to be both significantly improved and expanded within the community environment.

In the following sections, we focus on control-flow and data-flow design for lightweight process modeling.

4 Control Flow Modeling for Process-Oriented Mashups

In order to simplify business process modeling, models must be highly reusable, giving process flexibility and minimizing designs made from scratch. There is wide agreement that patterns can both accelerate the process of designing a solution and reduce modeling time [9]. Patterns enable participants of a community to communicate more effectively, with greater conciseness and less ambiguity [10], [11]. A pattern is an abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts [12]. The use of patterns is a proven practice in the context of programming, as shown by the impact made by the design patterns of Gamma et al. [13] as well as various other patterns.

We have adopted eight symbols from BPMN. Because BPMN has been well accepted by business users [14] and our research is focused on the design of an executable process modeling language not on the design of notations. This minimum number of notation means a user will have minimized learning efforts. Research shows that the average subset of BPMN used in process models consists of just nine different symbols [14]. Notation pool and lanes are dropped because they are used for expressing process collaborations. In the lightweight process modeling process collaboration can be expressed from an individual business partner perspective and an integrated view will be generated by the process editor. We have further adapted the notation for multiple instance business processes. The symbols adopted are presented in Fig. 1 Adopted Symbols from BPMN.

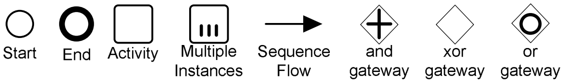

Fig. 1. Adopted Symbols from BPMN

We have also used the eight most frequently used workflow patterns (see Fig. 2) from [15] as our process patterns. The patterns can be captured within most common business process models and have a well-defined formal foundation.

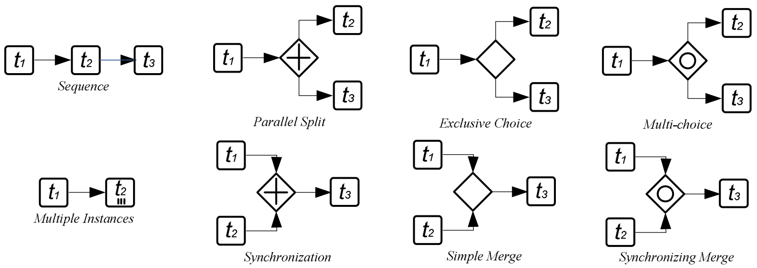

Fig. 2. Basic Process Patterns

These workflow patterns from [15] are however too fine-grained and have insufficiently information on the context and results to represent a reusable solution. Therefore, we introduce process templates that are different combinations of process patterns. The processes represented by a process template are sound. Certain process templates can be enriched with the information that they are valid for different domains, i.e. business context.

Using patterns, the soundness of process models can be guaranteed in certain ways [16]. The patterns can further be used to develop domain-specific process templates. Following context-aware modeling principles, it is possible to adapt the best fitting process artifacts during selections e.g., a user can fill in context information such as particular industry, location, and process name; a specific process template will appear to the users accordingly.

5 Data Flow Modeling for Process-Oriented Mashups

Data flow patterns aim to capture the various ways in which data is represented and utilized within processes. Russell et.al. present 40 data-related patterns for process-aware information systems [17] which are divided into four distinct groups: data visibility, data interaction, data transfer and data-based routing. Data visibility relates to the extent and manner in which data elements can be viewed by the various activities of a process. Data interaction focuses on the manner in which data is communicated between active elements within a workflow. Data transfer considers the means by which the actual transfer of data elements occurs between activities or sub-processes and describes the various mechanisms by which data elements can be passed across the interface of an activity or sub-process. Data-based routing characterizes the manner in which data elements can influence the operation of other aspects of the workflow, particularly from a control flow perspective.

See [18] for a discussion on data integration related issues of the current mashup tools. Data flow operators to be performed on either the structure of the data, or on the data itself. Data is generated and updated using different data refresh plans, such as pull and push strategies. The pull strategy is based on frequent and repetitive requests from the client; there are two ways to handle pull interval. In a global strategy, this will be set for the whole application. But in the local strategy, each data source is given its own refresh interval. In the push strategy, the client does not send requests but is required register with the server.

According to the definition of the data flow patterns in [17], the data operators and data refresh strategies in data-oriented mashups only cover data transfer, data interaction, and data routing. Data visibility is not covered by data-oriented mashups. The most popular data operators in the current data-oriented mashups are ``Union", ``Join", ``Sort" and ``Filter". A comparison of IBM mashup tool, Yahoo Pipes, Microsoft popfly and Google mashup editor is presented in [8].

From the business process application perspective, all data flow operators can be implemented using data transfer web services. In a process-oriented enterprise mashup, the data flow operators are not limited by the mashup providers. Collecting a library of web services which can support different data operators is useful and powerful for end users.

We have chosen 13 out of 40 dataflow patterns, i.e., three dataflow patterns for data visibility, five patterns for data interaction, three patterns for data transfer mechanisms, and two patterns for data-based routing. Multi-instance patterns are selected, which BPEL does not support. These 13 dataflow patterns provide an expressive power for the lightweight language. We do not expect that end users will be able to understand them completely when they first start to complete a mashup. However, we expect that they will be easily learned from existing process templates, reference models, and other process fragments.

6 Model Personal Collaborative Activity

To illustrate our idea we have modeled a personal collaborative activity that is often associated with social networks. A user named Lorraine uses data from three social websites to build her personal preferences about music, recipes, and book collections, as shown in Figure 3. Lorraine subscribes to RSS feeds from a popular music review website. She is only interested in piano pieces from Mozart. If any information in the

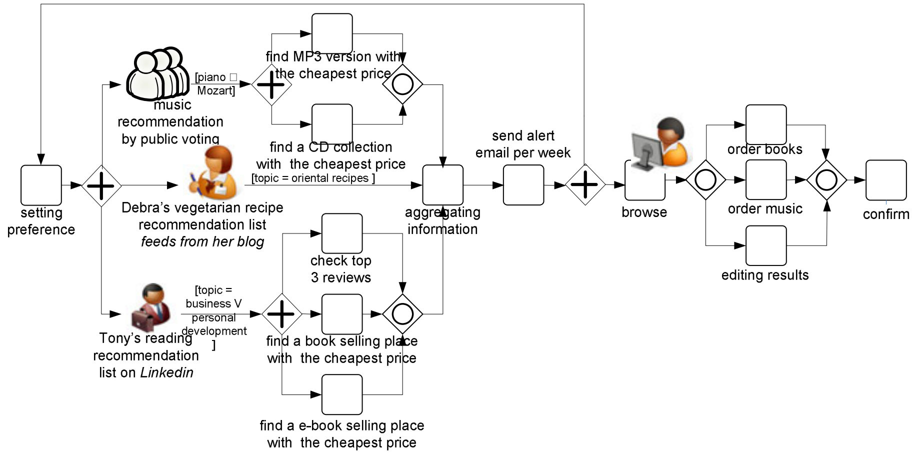
find MP3 version with the cheapest price
[piano □ Mozart]
music recommendation by public voting
find a CD collection with the cheapest price
send alert email per week
order books
setting preference
[topic = oriental recipes]
browse
order music
confirm
Debra's vegetarian recipe recommendation list *feeds from her blog*
aggregating information
editing results
check top 3 reviews
[topic = business V personal development]
Tony's reading recommendation list on *Linkedin*
find a book selling place with the cheapest price
find a e-book selling place with the cheapest price

Fig. 3. Personal Collaborative Activity Scenario

feed is related Mozart’s music, two web services are used to find the cheapest MP3 version and CD version from the Web. Lorraine also collects oriental vegetarian recipes from Debra's blog. Finally, she follows Tony's reading recommendation list on Linked-In. The topics of all books recommended by Tony on business or personal development will be selected. After receiving these book titles, Lorraine wants to read the top three reviews and find the cheapest price for the paperbacks, as well as any e-book versions.

All information related with music, recipes and books will be aggregated in the process. An alert email is sent to Lorraine every week. After Lorraine reads the aggregated information, she can decide whether to order books, music or save the results. She can also further trace the orders, etc.

7 Conclusions

Web-based technologies not only impact our communication patterns, but also provide opportunities to bring information and knowledge to our daily activities. Process composition in a process-oriented mashup provides an agile approach to adapting to changing business environments for virtual enterprises. This paper has presented the new concept of a process-oriented mashup. A lightweight process modeling method is provided for modeling the needs of a process-oriented enterprise mashup. The current version of the lightweight business process modeling language has eight graphical symbols and eight control flow patterns. A range of executable process templates are in used and development is continuing. Thirteen dataflow patterns have been chosen to support the current needs of perimeter enterprise applications. The primary evaluation of the lightweight process model method and language can be found from [7], [8].

Finally, we would like to emphasize that process-oriented mashups will not completely replace core business process management systems. Process-oriented mashup applications address different needs and are built for just a handful of users, applications that are used for only a few weeks or months, or situational applications that address a small piece of functionality. Process-mashups can serve users with different modeling skills for collaboration within a virtual enterprise environment.

Acknowledgement

The authors would like to thank the EU SOA4All and FAST projects. Part of work by Dr. Lai Xu and Dr. Paul de Vrieze was performed at the SAP Research, Switzerland.

References

1. Papazoglou, M.P., Georgakopoulos, D.: Service-oriented Computing. Communications of the ACM 46(10), 24–28 (2003)
2. de Vrieze, P., Xu, L., Bouguettayay, A., Yang, J., Chen, J.: Process-oriented Enterprise Mahsups. In: 2009 Workshops at the Grid and Pervasive Computing Conference 2009 Workshops at the Grid and Pervasive Computing Conference (2009)
3. de Vrieze, P., Xu, L., Xie, L.: Situational Enterprise Services. Encyclopedia of E-Business Development and Management in the Digital Economy. Idea Group Publishing (accepted)

4. Xie, L., de Vrieze, P., Xu, L.: When Social Software Meets Business Process Management. In: 2009 International Conference on Computer Sciences and Convergence Information Technology (ICCIT 2009), Seoul, Korea, November 24-26 (2009)
5. van der Aalst, W.M.P., Barthelmess, P., Ellis, C.A., Wainer, J.: Workflow Modeling using Proclets. In: Scheuermann, P., Etzion, O. (eds.) CoopIS 2000. LNCS, vol. 1901, pp. 198–209. Springer, Heidelberg (2000)
6. van der Aalst, W.M.P., Barthelmess, P., Ellis, C.A., Wainer, J.: Proclets: A Framework for Lightweight Interacting Workflow Processes. International Journal of Cooperative Information Systems 10(4), 443–482 (2001)
7. Xie, L., Xu, L., de Vrieze, P.: Lightweight Business Process Modelling. In: 2010 International Conference on E-Business and E-Government (ICEE 2010), Guangzhou, China, May 7-9 (2010)
8. Xie, L., Xu, L., de Vrieze, P.: Process Modelling in Process-oriented Enterprise Mashups. In: 2010 The 2nd IEEE International Conference on Information management and engineering (IEEE ICIME 2010), Chengdu, China, April 16-18 (2010)
9. Medicke, J., McDavid, D.: Patterns for Business Process Modeling. Business Integration Journal 1, 32–35 (2004)
10. Buschmann, F., Henney, K., Schmidt, D.C.: Past, Present, and Future Trends in Software Patterns. IEEE Software 24(7/8), 31–37 (2007)
11. Tran, H.N., Coulette, B., Dong, B.T.: Broadening the Use of Process Patterns for Modeling Processes. In: Proc. SEKE, Knowledge Systems Institute Graduate Schools, pp. 57–62 (2007)
12. Riehle, D., Zullighoven, H.: Understanding and using patterns in software development. Theory and Practice of Object Systems 2(1), 3–13 (1996)
13. Gamma, E., Helm, R., Johnson, R., Vlissides, J.: Design Patterns Elements of Reusable Object-Oriented Software. Addison-Wesley Publishing Company, Reading (1995)
14. zur Muehlen, M.: How much BPMN do you need? BPM Research, `http://www.bpm-research.com/2008/03/03/how-much-bpmn-do-you-need/`
15. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14(3), 5–51 (2003)
16. Gschwind, T., Koehler, J., Wong, J.: Applying Patterns during Business Process Modeling. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 4–19. Springer, Heidelberg (2008)
17. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Data Patterns. QUT Technical report, FIT-TR-2004-01, Queensland University of Technology, Brisbane (2004)
18. Di Lorenzo, G., Hacid, H., Paik, H.-y., Benatallah, B.: Data Integration in Mashups SIGMOD Record (2009)