Using Business Process Management Frameworks to Integrate Knowledge Management Services into Applications

Constantin Jucovschi
Jacobs University Bremen
DE-28757, Bremen, Germany
c.jucovschi@jacobs-university.de

Abstract

In [Dav+12], David et al. present the Semantic Alliance (Sally) — a framework enabling integration of Knowledge Management services into applications typically used by knowledge workers (e.g. spreadsheet programs). While integrating new applications with the Sally framework did not pose serious challenges, adding new Knowledge Management (KM) services was getting ever more challenging with every newly integrated service.

A closer look revealed that compared to other typical tools such as spreadsheet optimizers or rich visualization tools, Knowledge Management services typically have a very restricted set of software objects to which they can be applied. On the other hand, they can be applied in a much richer number of contexts. These requirements call for distinct integration strategies that this paper explores in more detail.

A solution to accommodate these requirements was implemented based on frameworks developed in the Business Process Management research area.

1 Introduction

In their paper [SL04], Stenmark and Lindgren present a collection of best practices for integrating Knowledge Management systems into everyday work. One such best practice is resisting the temptation of introducing new tools to the workplace of knowledge workers. The reason is that new tools undergo a long adoption process — time in which knowledge workers understand how the new tools integrate into their usual workflows. A better solution is to integrate KM tools into applications familiar and widely used by the user (also known as invasive technology [Koh05]). This is known to substantially increase the likelihood of survival of KM services and that the users will eventually discover the benefits of the added tools [SL04].

[Dav+12] presents the Semantic Alliance (Sally), a framework enabling integration of Knowledge Management services into applications typically used by knowledge workers (e.g. spreadsheet programs). The novel idea, coined in the paper as Invasive Design, was to combine invasive technologies with Semantic Illustration [KK09] (linking software objects e.g. spreadsheet cells to ontologies and thus providing fertile ground for KM services). To validate the framework, the authors created an implementation which integrated several KM services into Microsoft Excel and LibreOffice/OpenOffice. The KM services allowed the user to assign ontology concepts to blocks of cells and provided services like definition lookup and semantic navigation.

The initial implementation of the Sally framework proves the feasibility and usefulness of the invasive design paradigm. Subsequent efforts of adding new KM services to the already invaded spreadsheet systems showed that adding new services became ever harder because it was hard to specify when a certain service should be executed and how it should behave in response to events coming from the invaded system. The framework did not show any scalability problems at invading other applications like Autodesk Inventor (CAD System), jEdit (text editor), Planetary (web based framework for active documents). The KM service integration challenges were observed independently of the invaded system. These challenges substantially increased when developing a pricing service that used semantic data from both, a spreadsheet document and a CAD system to provide context-switching functionality from one system to the other [Koh+13]. One reason for this challenge was that services like definition lookup had to function in the same way independently whether the software object for which definition was requested was a CAD object or a spreadsheet cell.

The encountered scalability problems are not consequences of the Invasive Design paradigm. Invading new systems and linking software objects to an ontology did not pose difficulties. It merely provided a fruitful ground for KM services to come together. The real challenge was mashing up KM services together into one coherent system that the knowledge worker is willing to use.

In this paper, I would like to present the new extension mechanism of Semantic Alliance that partially alleviates the aforementioned problems by employing techniques from the field of Business Process Management (BPM). Even with the new extension mechanism, integrating new KM services into Sally might still require adaptations in configuration and even implementation of existing services. The number and difficulty of such adaptations is however reduced.

In the following section, I will shortly describe the main components of the Semantic Alliance framework and introduce important concepts used extensively for the rest of the paper. To give the reader a better idea of the encountered scalability problems, I describe in section 3 two services and show the problems that might occur if integrated into one system. In section 5, I will introduce a metamodel used in the area of Business Process Management to categorize existing process management frameworks. In section 4, I will describe in more detail the observations that were made during the efforts of mashing up KM tools. These will define the requirements an existing BPM framework would have to provide in order to be used for our purposes. Section 6, presents some detail about our implementation. The paper ends with a conclusion and a short discussion.

2 Sally Components

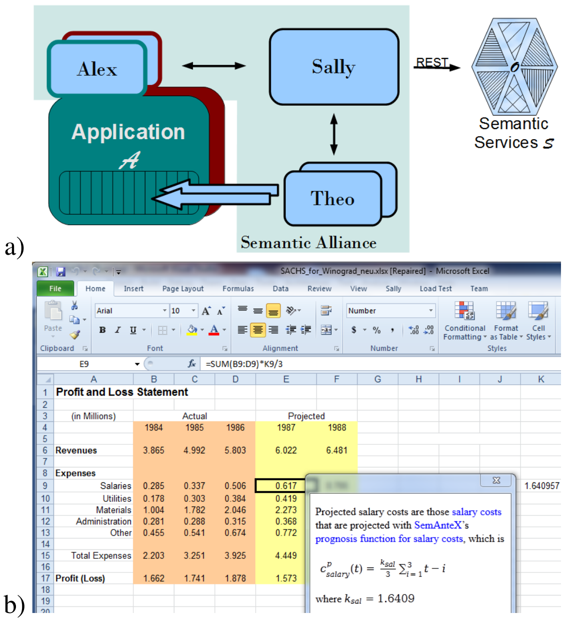

Figure 1: Sally Components while doing definition lookup

To explain the main components of the Sally framework, lets us look at Figure 1b). One can see an instance of Microsoft Excel and a window on top of it showing definition lookup for the cell E9. Microsoft Excel corresponds to Application $\mathcal{A}$ in Figure 1a) and is the system which was invaded. That means that Excel hosts a custom plugin (called an Alex[1]) which connects with the Semantic Ally component (Sally) and sporadically sends it information about the opened documents and user events. Based on the information and events sent by Alex, the Sally component may, at some point, decide that it wants to interact with the user of the invaded system. This is achieved through the Theo screen manager — a standalone program also connected to Sally and it is only job is to create windows anywhere on the screen (just like the definition lookup window) and load a certain web-page inside that window. In contrast to a normal browser window, the JavaScript loaded in a Theo window, can interact back with Sally and hence with Alex. Finally, semantic services like definition lookup, are implemented as standalone services possibly running on other servers and using completely different frameworks.

The original paper [Dav+12] introducing the Semantic Alliance framework discusses in great detail the reasons behind integrating KM services in such a way. It also provides the reader with a good intuition on the type of messages being exchanged between Alex, Sally, Theo and the semantic services.

[1]named after Alexander the Great; one of the mightiest invaders in history

3 Challenges in Mashing Up Knowledge Management Tools

The aim of this section is to make the reader familiar with the type of challenges one encounters when mashing up several Knowledge Management services.

The first service was already introduced in the previous section — the **definition lookup** service. The service works as follows:

1. The user makes Sally aware that she wants to perform definition lookup (e.g. clicking some menu item).
2. She clicks on a cell (e.g. E9) and if that cell has an ontology concept (e.g. "Projected salary costs") assigned to it,
3. Sally creates a Theo window showing the definition (Figure 1b).
4. Any following clicks on spreadsheet cells (e.g. E8), result in the Theo window to be moved to the right of the clicked cell and content of the definition lookup window to be updated.

The second service, called **ontology linking service**, allows the user to assign ontology concepts to blocks of cells. Here is a description of the service.

1. The user makes Sally aware that she wants to perform ontology linking (e.g. clicking some menu item).
2. She clicks on a range of cells (possible just one cell), and if that range has:

3a. no ontology concept assigned to it, then Sally creates a Theo window showing a form that the user needs to complete and so assign an ontology concept to that range of cells. If she selects another range of cells, only one element in the forms gets updated, namely the one showing the range for which the ontology link will be assigned.

3b. has an ontology concept assigned to it, a form with previously saved information is shown to the user allowing editing. If the user does not change the form but selects a new range, Sally closes the current Theo window and goes to step 2, otherwise, updates the range field of the opened form.

Both services have some relatively simple way of responding to events coming from the invaded spreadsheet system. Nevertheless, mashing them up in one system is not trivial. Imagine the user requests ontology linking service, selects a range of cells without a link to an ontology concept and starts filling in the form. In the middle of the task, she feels the need to do definition lookup on some other cell and so she selects it and invokes the definition lookup service.

At this point the Sally component is requested to run two services in parallel and there are several ways it can respond to that. Sally could implement a "one service at a time" policy which would frustrate the user. Sally could also let both services run in parallel and forward all incoming events to each service. That would eventually confuse the user because her action would have consequences in services that for her mental model are not in focus / not active. Hence Sally needs to have an event forwarding strategy which would forward events only to one service — the one for which the event was intended.

The initial implementation of the Semantic Alliance framework used state variables to control which components would be notified of incoming events. As the number of services grew, it became very hard to manage such forwarding rules. In particular, to integrate a new service, changes in several related services had to be performed and hence modularity was violated. Additionally, in order to customize this behavior, the whole framework had to be recompiled.

4 Mashing up Semantic Illustration Services

The Semantic Illustration (SI) architecture [KK09] enhances an application with the Interpretation Mapping function IM which assigns each software object a concept in an ontology $\mathcal{O}$. The ontology $\mathcal{O}$ formalizes relationships among concepts and possibly connects them to external ontologies. Note that the ontology $\mathcal{O}$ is not static. As the user changes or opens new documents, the ontology $\mathcal{O}$ gets updated/extended with new concepts and relationships.

Semantic Illustration Services heavily tap into the information provided by the interpretation mapping and hence it is worthwhile to analyze how the properties of the IM function affects the type of interactions Semantic Illustration Services might be able to provide.

4.1 Dynamic Applicability

Semantic Illustration Services usually have very specific requirements that need to be satisfied before they can be applied. For instance the definition lookup service requires that the input software object has a link to an ontology concept which in turn has a definition. A complex service like the pricing service in [Koh+13] requires that the input CAD object cad satisfies

$$\begin{aligned} \exists cell1, cell2. & \quad \text{such that} \\ IM(cell1) &= IM(cad) \wedge \\ IM(cell2) &= \text{"http://economics.org/prices"} \wedge \\ costOf(cell1, cell2) &\in \mathcal{O} \end{aligned}$$

As documents change or new documents are opened, Semantic Illustration services can become applicable or cease to be applicable.

The requirements that need to be satisfied before a Semantic Illustration service can be invoked may become extremely convoluted and should be regarded by a mashup framework as black boxes. The mashup framework should provide a mechanism for the user to get the list of services are applicable in the current context.

4.2 High Reusability

An important observation about the interpretation mapping IM is that it does not depend on any context information. Hence independently of whether spreadsheet cell $C4$ is used as part of a formula or as parameter in a wizard, the $IM(C4)$ is always the same. As a consequence, Semantic Illustration services that solely use information provided by the interpretation mapping are context independent and can be invoked from a multitude of contexts.

The definition lookup service is an example of such a service as it solely uses the IM function to fetch the definition of the software object at hand and display it. Indeed, the user should be able to invoke the definition lookup service for cell $C4$ in many situations e.g. if she: 1) clicks on cell $C4$ inside the spreadsheet document; 2) inspects a reference of cell $C4$ as part of a formula; 3) uses a wizard or plugin to perform operations on cell $C4$. Moreover, it may very well happen that a knowledge worker wants to get a definition lookup on a concept mentioned in the definiens provided by the content produced by the definition lookup service itself.

Due to the multitude of contexts in which a Semantic Illustration service can be applied as well as due to modal windows, it is often unfeasible to provide fixed user interface components for accessing the service. Context menus generally provide a better solution. It is the responsibility of the invading Alex component to reliably detect the software objects the user selects. This, in turn, depends a lot on the architecture of the invaded system.

4.3 Support for Subtasking

Knowledge Management tasks often require the user to perform actions she is not familiar with e.g. creating ontology concepts and making their relationship with other concepts explicit. In such cases, the knowledge worker would be tempted to look how similar tasks were performed before and complete current task by following the example of another one. That naturally introduces subtasking i.e. starting a new task while in the midst of another task.

In the example described in section 3, I have presented the challenges associated with the use case when two services run simultaneously and compete for in the same software object selection event. Such situations are quite common for Semantic Illustration services due to the fact that IM function needs a software object as parameter.

A framework for mashing up Semantic Illustration needs to have a policy for broadcasting events to SI services which does not confuse the user. Moreover, there should be an intuitive way how the knowledge worker can resume an interrupted service.

4.4 Access to Common Resources

KM services often rely on other KM services to achieve their goals. For example, the IM editor, a service linking software objects to ontologies needs to rely on some service that provides CRUD (Create, Read, Update, Delete) operations on ontologies. Our implementation uses the Planetary [Koh12] system for that, but there is no reason why one could not connect use Protege [Pro] or Semantic Media Wiki [Sem].

Similarly, in our experiments invading Web environments, we needed to make sure that services don't use the desktop version of the Theo screen-manager but the Web-based one.

It is desirable that KM services depend on each other because that is the key to reusability. The dependencies should be loose, so that one can easily exchange used implementations based on the context.

5 Dimensions of Business Process Modeling

The area of Business Process Modeling aims at describing actions, agents, resources, and relations among them necessary to complete a certain task. Due to the practical nature of the problem and applicability for business, there is a large body of research, best practices, standards, and implementations available. Most of these frameworks support basic modeling tasks but handle more complex situations in some particular way. This makes it very difficult to decide whether a framework is suitable for particular needs without becoming an expert in it. To partially alleviate this problem [LK06] developed a generic meta-model, i.e., a set of dimensions that can be used to categorize particular frameworks. In this section I provide a short description of

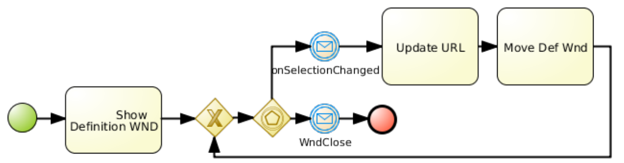

Figure 2: Definition Lookup

these dimensions. In section 5.1, I will evaluate how the mashup requirements from section 4 can be supported by Process Modeling Tools.

Functional Dimension captures the actionable elements of a process and the flows of data relevant to them [CKO92]. It is represented as a directed graph where the nodes represent the actionable elements and the edges represent the type of objects these elements need to exchange. Frameworks are classified in 3 categories, those which: do not represent activities (e.g. Petri Nets); only consider atomic activities; and those which allow subprocesses divisible into other subprocesses and atomic activities.

Behavioral Dimension describes when process elements get executed. This dimension captures loops, branching conditions, decision making, exit criteria etc. Frameworks are analyzed in respect with types of control flows they support e.g. AND splits/joins, XOR splits/joins, N-out-of-M joins etc.

Organizational Dimension captures where and by which agents process actions will be executed. Categorization criteria is based on the types of agents that can be modeled in the framework. For example, is role based execution possible? Can one differentiate between computer services and human workers?

Informational Dimension describes the types of entities that are exchanged among and changed by activities. Such entities include: Events, Database Tables but also Services, Applications.

Context Dimension presents a high-level view on the business process. It describes goals, their measures, deliverables, process owners, process types etc. It captures relations between processes (e.g. support processes); what deliverables are created and what goals these achieve.

5.1 Evaluation of Requirements for SI Service Mashups

The minimum requirements a Business Process Modeling framework needs to support for mashing up SI Services depend on two factors:

- the individual requirements of each SI Service towards the BPM framework.
- requirements to support mashing up SI services.

This section evaluates only the later.

An observation that makes modeling of dynamic applicability, high reusability and subtasking requirements much easier is that they can run in a process separate from the SI services themselves. To illustrate this, let us look at Figure 2 depicting the BPM diagram of the definition lookup service. The diagram does not contain any special actions or events that would allow for dynamic applicability or high reuseability. The later are enabled through the

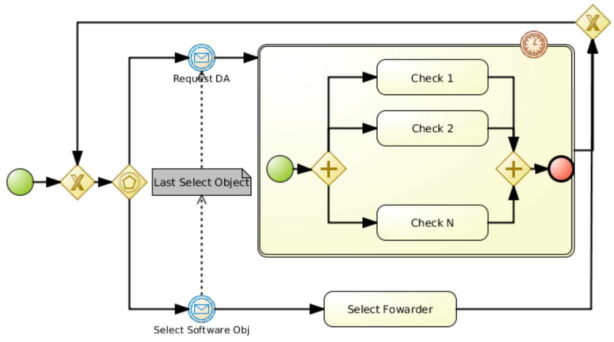

Figure 3: The EventLoop Process

EventLoop process in Figure 3. This process is always active and is the only one which directly receives events about selection events of software object ("Select Software Obj" event) or that the user requested a list of services applicable to the current context ("Request DA"). In case a new software object was selected, it runs a task ("Select Forwarder") which uses some heuristics to decide which service (e.g. definition lookup) should be notified of the event and forward the event to that particular service. This is how "onSelectionChanged" event from Figure 2 gets triggered.

In case the user wants to see what services are available for the current context, the engine starts a subprocess in which it runs in parallel a series of N service applicability checks, optionally have a timeout strategy and a way to create a list of services for which applicability checks succeeded.

The proposed solution gives us the following requirements for BPM frameworks:

- support for subprocesses for the functional dimension
- support for AND joins/splits for the behavioral dimension
- no special requirements for the context and informational dimensions

6 Implementation

To implement the extension mechanism for the Semantic Alliance, I used the jBPM Business Process Management suite [Jbp]. It complied with all the requirements presented in section 4 and also is Java based can so could easily be integrated with the existing Semantic Alliance implementation.

The implemented extension mechanism is based on distributing jar files. On startup, Sally loads all the jar files from a fixed directory and uses Java reflections mechanisms to find all BPMN2 process files. Each BPMN2 process file represents a Knowledge Management service. Each task inside such a process file, specifies an implementation class and method that should be called as soon as the workflow reaches that element. Such BPMN2 files can be created using the jBoss Eclipse plugin or using the web interface of jBoss Drools Guvnor.

7 Conclusions

This paper tackles the problem of mashing up Knowledge Management services in a modular way, in systems typically used by knowledge workers. The aim is to create a framework which could provide plug and play support for KM services. As described in section 4, Knowledge Management services have different integration requirements into applications than software artifacts such as optimizers or rich visualization tools. In comparison to these tools, Knowledge Management services typically have a very restricted set of software objects to which they can be applied. On the other hand, they can be applied in a much richer number of contexts. Supporting such an integration strategy is the main challenge and contribution of this paper.

The solution presented in the current paper is based on the very mature Business Process Management research area which aims at describing actions, agents, resources and relations among them necessary to complete a certain task. It seemed to provide a natural solution to our challenges and yet the presented solution does not cover all the requirements we identified. Further research is necessary to see if the presented architecture could be improved.

Another research area that could provide some useful insights is the one of ubiquitous computing. This area has similar challenges of making heterogeneous systems work together and respond to user actions in way that makes sense. The author found several goal-based frameworks which have the potential of vastly improving presented architecture.

The Semantic Alliance framework [Dav+12] and the challenges encountered in extending it with new Knowledge Management services, was used both as a tool to understand the root the integration challenges as well as a testbed for developing solutions. An experimental implementation of the extension mechanism was implemented and we are still in the process converting existing KM services to the extension mechanism described in this paper.

8 Acknowledgements

I want to thank Andrea Kohlhase and Michael Kohlhase as well as the anonymous reviewers for their constructive suggestions. This work has been funded by the German Research Council under grant KO-2484-12-1.

References

[CKO92] Bill Curtis, Marc I. Kellner, and Jim Over. "Process modeling". In: *Commun. ACM* 35.9 (Sept. 1992), pp. 75–90. ISSN: 0001-0782. DOI: 10.1145/130994.130998. URL: http://doi.acm.org/10.1145/130994.130998.

[Dav+12] Catalin David et al. "Semantic Alliance: A Framework for Semantic Allies". In: *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics (CICM) (Bremen, Germany, July 9–14, 2012). Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 49–64. ISBN: 978-3-642-31373-8. URL: http://kwarc.info/kohlhase/papers/mkm12-SAlly.pdf.

[Dav12] Catalin David. "Semantic Alliance Framework: Integrating Documents and Semantic Services". M. Sc. Thesis. Jacobs University Bremen, 2012. URL: https://svn.eecs.jacobs-university.de/svn/eecs/archive/msc-2012/cdavid.pdf.

[Jbp] *jBPM Business Process Management (BPM) Suite from jBoss Community*. URL: http://www.jboss.org/jbpm (visited on 07/15/2013).

[Jeu+12] Johan Jeuring et al., eds. *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics (CICM) (Bremen, Germany, July 9–14, 2012). LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012. ISBN: 978-3-642-31373-8.

[KK09] Andrea Kohlhase and Michael Kohlhase. "Semantic Transparency in User Assistance Systems". In: *Proceedings of the 27th annual ACM international conference on Design of communication (SIGDOC)*. (Bloomington, Indiana, USA, 2009). Ed. by Brad Mehlenbacher et al. ACM Special Interest Group for Design of Communication. New York, NY, USA: ACM Press, 2009, pp. 89–96. DOI: 10.1145/1621995.1622013. URL: http://kwarc.info/kohlhase/papers/sigdoc09-semtrans.pdf.

[Koh+13] Andrea Kohlhase et al. "Full Semantic Transparency". In: *Proceedings of the 14th IFIP TC 14 International Conference on Human-Computer Interaction*. INTERACT'13. in press. Cape Town, South Africa: Springer-Verlag, 2013. URL: http://kwarc.info/kohlhase/papers/Interact2013_FST.pdf.

[Koh05] Andrea Kohlhase. "Overcoming Proprietary Hurdles: CPoint as Invasive Editor". In: *Open Source for Education in Europe: Research and Practise*. Ed. by Fred de Vries et al. Proceedings at http://hdl.handle.net/1820/483. Open Universiteit Nederland. Heerlen, The Netherlands: Open Universiteit Nederland, Nov. 2005, pp. 51–56. URL: http://hdl.handle.net/1820/483.

[Koh12] Michael Kohlhase. "The Planetary Project: Towards eMath3.0". In: *Intelligent Computer Mathematics*. Conferences on Intelligent Computer Mathematics (CICM) (Bremen, Germany, July 9–14, 2012). Ed. by Johan Jeuring et al. LNAI 7362. Berlin and Heidelberg: Springer Verlag, 2012, pp. 448–452. ISBN: 978-3-642-31373-8. arXiv: 1206.5048 [cs.DL].

[LK06] Beate List and Birgit Korherr. "An evaluation of conceptual business process modelling languages". In: *Proceedings of the 2006 ACM symposium on Applied computing*. SAC '06. Dijon, France: ACM, 2006, pp. 1532–1539. ISBN: 1-59593-108-2. DOI: 10.1145/1141277.1141633. URL: http://doi.acm.org/10.1145/1141277.1141633.

[Pro] *Protégé*. URL: http://protege.stanford.edu.

[Sem] *Semantic MediaWiki*. URL: http://semantic-mediawiki.org (visited on 03/04/2010).

[SL04] Dick Stenmark and Rikard Lindgren. "Integrating knowledge management systems with everyday work: Design principles leveraging user practice". In: *System Sciences, 2004. Proceedings of the 37th Annual Hawaii International Conference on*. IEEE. 2004, 9–pp.