A Top-Down Approach Based on Business Patterns for Web Information Systems Design

Marinette Savonnet, Jean-Claude Simon,
Marie-Noëlle Terrasse, and Éric Leclercq

University of Burgundy
Le2I Laboratory - UMR 5158
B.P. 47 870, 21078 Dijon Cedex - France
{Marinette.Savonnet,Jean-Claude.Simon,Marie-Noelle.Terrasse,Eric.Leclercq}@u-bourgogne.fr

Abstract. In this paper we develop an approach that is based on a top-down strategy to realization of transactional web services. Our approach highlights non-functional properties (e.g., traceability, security) which are essential to preserving an application's quality. It is implemented in three steps. The first step is a breakdown of the application in accordance with a related business involved. The goal of this step is to have sets of actors and activity patterns defined as an activity workflow that support the architecture of the application. The next step allows developing a mapping of the activity pattern on this architecture. The aim of this step is to identify the risk for non-functional properties. The last step is the translation of patterns in abstract web service.

Keywords: eco-system, non-functional properties, transactional web service

1 Introduction

As discussed in a European report [1], the ISU (Interoperability Service Utility) layer would allow enterprise interoperability in the "Service Oriented" world. ISU should provide enterprises with modular software building blocks corresponding to a functional decentralization of their business areas. One of the major issues here is the fact that software block combinations do not scale up easily in terms of their complexity, their validation, their compensation mechanisms, etc. Usually software building blocks are composed (e.g., by using choreography and orchestration of web services) in a bottom-up manner in order to provide complex web services. The European report [1] also emphasizes the importance of assuring certain properties in the context of the eco-system of enterprises.

In this paper we propose an approach for Web Information Systems design. This approach is based on a top-down strategy that focuses on a business-driven analysis of an eco-system's activities in order to determine combinations of software blocks. One of the major issues dealt with in this paper is how to reveal all interactions between software blocks that are tied to non-functional, yet essential, properties such as traceability, data confidentiality, and security. We then combine existing web services in order to build a high-level technological framework which would allow to develop our application. Essential non-functional properties are used as guidelines in order to facilitate convergence of the functional and technological aspects.

Section 2 presents context of our example and sections 3 to 5 outline our approach. Section 6 presents our on-going work.

2 Context of our application

We apply our approach to the DEFOR[1] application example. The network for the sustainable management of cultivated forest aims to develop collaborations between the research world and the forestry sector in Western Europe. These collaborations are encouraged by the project DEFOR. This pilot-project will serve as a basis for developing an information system intended to small and medium-sized enterprises of the forestry sector. The goal of the DEFOR project is to develop a web platform for collecting, archiving, processing and exchanging of data among forestry professionals (Figure 1).

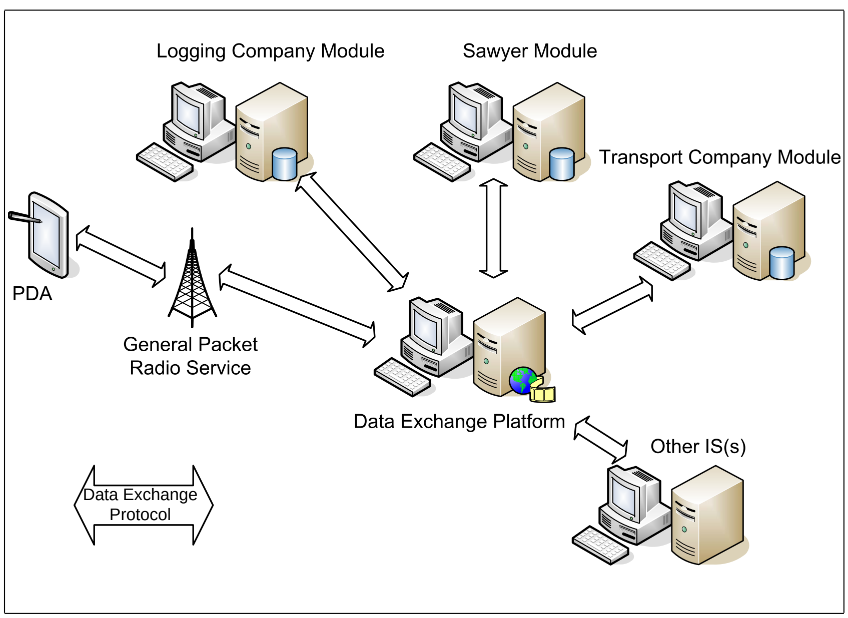

Fig. 1. The DEFOR platform

They are three types of applications that allow to use the DEFOR platform:

[1] DEFOR standing for "DEveloppement FORestier", i.e., Forestry Development

1. thin-client applications installed on a PDA and connected to the Internet;
2. thin-client applications installed on the server and invoked from a PC connected to the Internet via a web browser;
3. thick-client applications installed on the user computer and using data stored on an external server by using an Internet data exchange protocol.

The timber logging field is represented by a set of actors, each having a specific role. First, **forest owners** generally sell their timber still standing to the **logging companies**. These companies organize the work of cutting down trees and dragging them to road-sides, where they are stored in stacks. The transport of timber is carried out by **transport companies**. Their vehicles are special trucks equipped by forestry cranes. Their **drivers** should have a dual qualification, namely driving trucks and operating their cranes. Their **logisticians** organize transport of timber. The destinations are the sites of wood processing, i.e., the **customers**.

The Use Case diagram in Figure 2 describes actors and main functionalities of the DEFOR project.

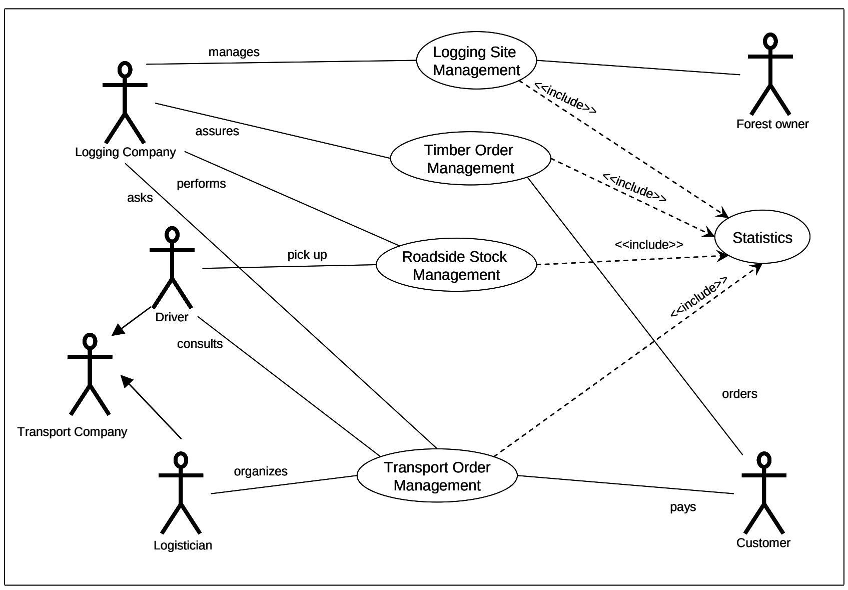

Fig. 2. The DEFOR Use Case diagram

We propose a methodology in three steps:

1. definition of an architecture that will support the application;
2. defining and mapping patterns of activities onto this architecture and detecting "points of risk" to the guarantee of non-functional properties;

3. and setting up the corresponding web services.

We will follow these three steps in the sections below.

3 Definition of an architecture for supporting eco-system interactions

In this first step, we are interested in transforming the target application into an organization structure relevant to the business area concerned. In order to elaborate an appropriate business architecture, we ask three questions.

3.1 What are the parties involved?

This first question pertains to definition of roles and groups analogous to Van Der Aalst's organization step [2]. We strive to define abstract eco-system's actors, depending on both competencies and affiliations. We are talking about actors in the sense of UML case diagrams. This work needs to be done at a high level of abstraction and also needs to take into account differences, when certain persons play different roles in different structures.

For the DEFOR project, we identified six actors and four main functionalities (Figure 2) from the requirement specification.

3.2 What are the essential non-functional requirements of the system?

This second question pertains to definition of non-functional properties that must be guaranteed by the eco-system. We are interested in requirements that are strong enough to result into changes of the eco-system's business architecture. Our objective is to find and then to eliminate all risks that are due to the interaction of various information systems that preserve different non-functional properties [3]. Certain non-functional properties are essential to business processes quality and are highly specific to a given application domain (e.g., traceability, security). Such properties must be guaranteed even in the context of applications that are developed by using complex web services.

It appears at this point that these needs are identified through discussions with the future users of the application. The main task is to identify relevant non-functional aspects and also to identify the essential needs of the users.

In the DEFOR project, the **timber industries** have specific requirements: They want to be able to buy timber conforming to specifications for the quantities and time schedules. The corresponding non-functional property is called the *obligation of result.*

3.3 What is the generic activity pattern that is characteristic of each actor or a group of actors?

This third question pertains to definition of a generic activity pattern that could describe all of the eco-system activities. An activity pattern is a combination of activities having an objective, a logical sequence, and dependencies implied by business rules.

In order to answer this third question, we use a block-oriented strategy that is similar to that used by Schimm [4] for process mining. We build activity pattern blocks that belong to core business areas of an eco-system's actors.

Various research projects have been conducted on the decomposition of business processes: a survey of the state of the art can be found in [2]. We structure our patterns analogously to Dustdar & al. [5]. These authors refer to it as the level of abstraction workflows in their presentation of WSIM (Web Services Interaction Mining). The basic mechanisms that we model are based on twenty business patterns of Van Der Aalst & al. [6]. Then we can follow the method proposed by White [7] for transformations of models among the patterns of Van Der Aalst, UML activity diagrams, and the BPMN notation.

We wish to decompose the DEFOR project into generic functionalities. Since the DEFOR project is specific to the timber business, we find in it the usual commerce activities. We thus find software modules related to ordering, order preparing, pick-up and delivery of timber, and finally invoicing. These activities are followed by a sequence of ordered tasks, which are represented by a pattern of high-level activities (Figure 3).

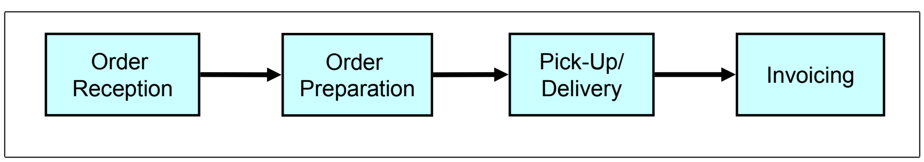

Fig. 3. Pattern of high-level activities

At the end of this work step, we obtain a set of actors and a pattern of activity defined by a workflow of activities and by transitions between the activities.

4 Mapping activity patterns onto the business architecture and detecting "points of risk" to the guarantee of non-functional properties

This decomposition will be carried out along two axes: an **activities** axis (i.e., horizontal axis), and a **business** axis (i.e., vertical axis). The activities axis describes the sequence of, possibly complex, activities for which the information system is to be built. The business axis allows to divide such activities into several specialties: the core business area, and the associated business areas.

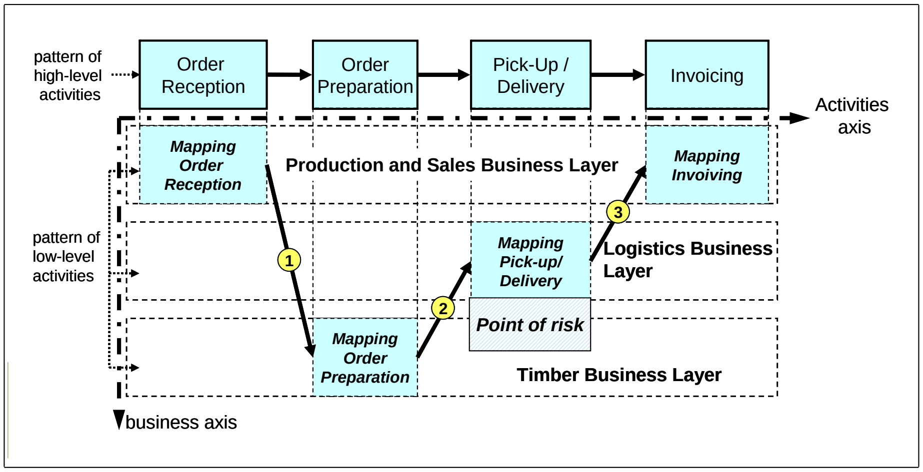

Fig. 4. Mapping of high-level activities into business layers

For the DEFOR project, we add three layers into the pattern of high-level activities:

- Production and Sales Business Layer,
- the Logistics Business Layer,
- and Timber Business Layer.

The *Production and Sales Business Layer* corresponds to companies that produce and sell goods. The *Timber Business Layer* represents the core business area in the DEFOR project. The *Logistics Business Layer* is intended for logistics companies that organize transport and delivery of products.

In our project, the ordering and invoicing activities are projected into the *Production and Sales Business Layer*, order preparation activity is located in the *Timber Business Layer*, and finally pick-up and delivery activity is located in the *Logistics Business Layer*. Figure 4 illustrates the mapping of activity pattern onto the above three layers.

Any transition that crosses layer boundaries is a potential "point of risk" to the guarantee of non-functional properties. The objective is to build layers containing sub-patterns of activities that do not include points of risk that would jeopardize non-functional properties.

For DEFOR project, one problem to be handled concerns the management of stacks of the road-side stock: The driver cannot find the wood stacks at the expected place where he should pick up a part of his cargo. This implies, e.g., a stack of wood that is not in the place indicated on the map (this may result from an input error or an offset between the GPS tracking of trucks and the driver's instructions). During the follow-up, the logistician will consult maps with additional functionality. These consultation will allow to access alternative information on location of pick-up places, delivery routes, etc. It is the role of logistician to resolve various problems encountered by drivers. This may possibly modify the initial preparation of an order (see figure 4-Point of risk).

In addition, fine-grain patterns are also described in order to give more details on each of the coarse-grain activities. We obtain three layers that contain activity patterns of different granularity. We stop refining patterns of activities when the activities can be described by using only 1) activities that, together with their control, are internal to a single information system, and 2) control constructs that appear in Van der Aalst classification.

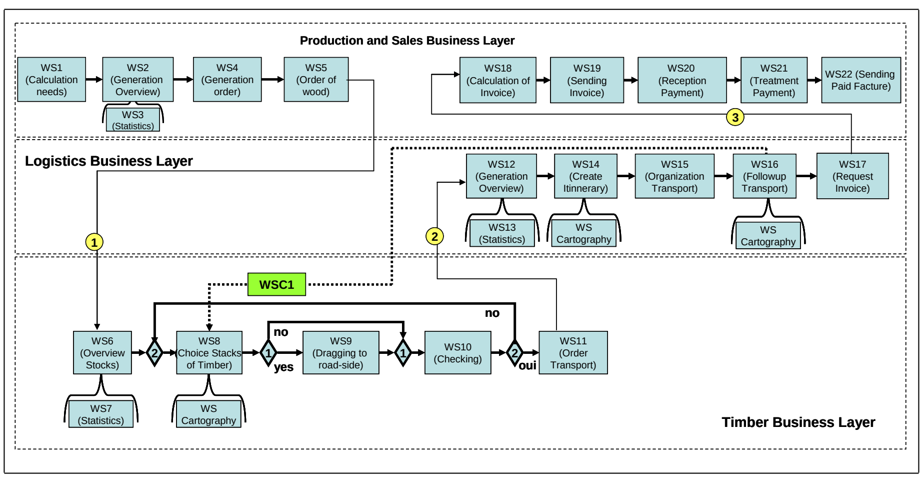

Fig. 5. Web services for DEFOR project

5 Defining abstract services

We use Dustdar & al. [5] definition of abstract services, i.e., web services corresponding to activity blocks that describe business processes. Such a definition proceeds as follows:

- Each activity of the fine-grain patterns is associated with an abstract web service. Such a web service is called an action web service.

- Each action web service is attached to an activity that is a source of a transition marked as risky to non-functional properties is associated with an external web service for compensation. Such a web service for compensation is in charge of guaranteeing non-functional properties while the effect of its corresponding action web service is reversed. In such a case, the action web service is a pivot web service, as defined by Bhiri [8], i.e., the standard compensation mechanism does not apply to this action web service.
- Each abstract web service which is not associated with a risky activity is realized by a combination of web services. On such combinations, the standard compensation mechanism is applied without restrictions.
- The complex activity pattern thus produced includes of transitions of different nature: the classical transitions from one activity to another, and the transitions of recovery which correspond to the invocation of compensation web services.

In figure 5, the WS16 web service has compensation web service associated to it. This compensation web service, called WSC1, manages transitions between the Logistics and Timber Business Layers. It is these transitions that may jeopardize the property of obligation of result.

6 Conclusion

In this paper we have proposed a top-down business-oriented approach for defining patterns. The resulting patterns are further refined in order to include increasingly specific features of an enterprise's core business. In order to satisfy non-functional requirements, we have to accurately identify activity flows that cross pattern boundaries.

An example implementation with web services is presented. Our implementation uses compensation web services that can guarantee non-functional properties.

In order to be able to improve quality of our analysis of business processes of an eco-system of enterprises, we plan to use process mining methodologies. The first such proposal was that of Van Der Aalst [2]. Another approach is to look for autonomous blocks of activity in business processes. We plan to apply such an alternative approach to future case studies of enterprises whose core business influences all of their business processes.

In both of the above cases, we need to extend existing notations (e.g., UML activity diagrams, BPMN) in order to describe risky activities and transitions. It will be also necessary to extend White's transformations between UML and BPMN notations so that transformation rules between UML activity diagram's extensions and BPMN's extensions can be properly defined.

Our future work on mappings of activity patterns into a business architecture will propose additional criteria for evaluation of mapping quality. Such criteria should take into account preservation of non-functional properties. Such criteria should also guarantee adequacy of mappings with respect to information systems that are implemented in the core business rules. By using such evaluation criteria we hope to be able to define a strategy that could be applied when several non-functional properties are required, possibly leading to low-quality mappings.

References

1. Li, M.-S., Cabral, R., Doumeingts, G., Popplewell, K.: Enterprise Interoperability – Research Roadmap – Version 4.0. Technical report, European Community, Information Society Technologies `ftp://ftp.cordis.europa.eu/pub/ist/docs/directorate_d/ebusiness/ei-roadmap-final_en.pdf` (2006)
2. van der Aalst, W., Weijters, A.: Process Mining: a Research Agenda. Computers in Industry 53(3):231-244 (2004)
3. Duarte-Amaya, H.: Tcows Canevas pour la Composition de Services Web avec Propriétés Transactionnelles. Phd, Université J. Fourier, Grenoble (2007)
4. Schimm, G.: Generic Linear Business Process Modeling. In: ER '00: Proceedings of the Workshops on Conceptual Modeling Approaches for E-Business and The World Wide Web and Conceptual Modeling, LNCS, pp. 31–39. Springer (2000)
5. Dustdar, S., Gombotz, R.: Discovering Web Service Workflows using Web Services Interaction Mining. Int. J. Business Integration and Management 1(4):256–266 (2006)
6. van der Aalst, W., ter Hofstede, A., Kiepuszewski, B., Barros, A.: Workflow Patterns. Distributed and Parallel Database 14(3):5–51 (2003)
7. White, S.: Process Modeling Notations and Workflow Patterns. In: Workflow Handbook, L. Fischer (ed.) pp. 265–294. Future Stragies Inc. (2004)
8. Bhiri, S., Godart, C., Perrin, O.: Transactional Patterns for Reliable Web Services Compositions. In: ICWE, pp. 137–144. ACM (2006)