



HAL
open science

Visualizing and Assessing a Compositional Approach of Business Process Design

Sébastien Mosser, Alexandre Bergel, Mireille Blay-Fornarino

► **To cite this version:**

Sébastien Mosser, Alexandre Bergel, Mireille Blay-Fornarino. Visualizing and Assessing a Compositional Approach of Business Process Design. Software Composition 2010, Jun 2010, Malaga, Spain. Springer's Lecture Notes in Computer Science. hal-00531032

HAL Id: hal-00531032

<https://hal.archives-ouvertes.fr/hal-00531032>

Submitted on 1 Nov 2010

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Visualizing and Assessing a Compositional Approach of Business Process Design

Sebastien Mosser¹, Alexandre Bergel², Mireille Blay-Fornarino¹

¹ University of Nice Sophia – Antipolis
CNRS, I3S Laboratory, MODALIS team
Sophia Antipolis, France

{mosser,blay}@polytech.unice.fr
²Department of Computer Science (DCC)
University of Chile, Santiago, Chile
www.bergel.eu

Abstract. In the context of Services Oriented Architecture (SOA), complex systems are realized through the design of *business-driven processes*. Since the design of a complete process can be very complex, composition tools such as *aspects* and *features* propose to define large systems by composing smaller artifacts (more easy to understand) into a complex one. But these techniques shift the system complexity into the definition of composition directives able to build it. At composition time, process designers needs to be supported to understand and assess their designed systems. We propose in this article a set of visualizations to represents *compositions* of business-processes and then identify patterns and categorization. We use the ADORE framework as underlying process composition platform. We validate this work by presenting in this article instance of these visualizations obtained from a Car Crash Crisis Management system (CCCMS, a comparison referential for Aspect Oriented Modeling techniques). We use these visualization to assess the CCCMS realization.

Note for the proceeding reader: this paper makes use of colors. Although not mandatory for its understanding, an online (colored) version of this paper will ease the reading.

1 Introduction

An application that follows the Service Oriented Architecture paradigm (SOA, [?]) is an assembly of services that realizes business processes. Business processes are defined by business specialists and typically involve many services that are composed in a variety of ways. Furthermore, the need to extend a SOA application with new business features (to follow market trends) arises often in practice. In the technological context of Web Services, business processes can be implemented as *orchestrations of services* [?]. Existing tools and formalisms related to business processes (*e.g.* BPMN notation [?], BPEL industrial language [?]) use a

design-in-the-large approach and do not intrinsically provide language constructions and frameworks to support the introduction of new features into existing processes. Using these technology, one can define “*composition of services*” to reify business-process and then visualize such entities [?].

In this paper, we address the “*composition of process*” problematic, *i.e.*, the composition of existing composition of service to obtain more complex process at the end. New paradigms such as aspects [?] and features [?] model application in terms of composing smaller units. Assuming that a complex system is difficult to understand by humans, they propose to reduce the complexity by defining several smaller artifacts instead of a single and large one. They identify and encapsulate parts of models that are relevant to a particular concern. A same feature may be shared and integrated into several processes simultaneously. These artifacts are then composed to produce the expected system. These approaches help taming the complexity of business processes design [?,?]. As a consequence, the intrinsic complexity of the system is shifted into the composition directives used to build it. When a system involves many processes, it is necessary to have a holistic point of view on features, compositions and business processes to grasp it. In this paper we examine visualization method to tackle complexity of compositions at the application level.

The paper makes the following contributions and innovations:

- a number of visualizations dedicated to support designers when they define business processes using a compositional approach.
- benefits and design weakness are revealed using a number of patterns to assess compositions quality.
- scalability of the approach is sketched by using a very large case study as running example.

We motivate this contribution by presenting in section 2 our running example. Visualizations used to represent and assess compositions are then described in Section ???. Section ??? briefly presents implementation details. We propose a discussion on the approach benefits (associated to interesting perspectives) in Section ???. Finally, Section ??? describes an overview of related work, and Section ??? concludes this paper.

2 Running Example: Realizing a *CCCMS* using ADORE

This section presents the running example used to validate the visualization approach defended in this paper. It also presents the composition framework we used to realize this example, and highlights our identified needs of visualization.

2.1 Case Study: A Car Crash Crisis Management System

In Kienzle *et al.* [?], authors propose a common case study (a Crisis Management System, CMS) to compare existing *Aspect Oriented Modeling* approaches between each other. We consider this case study as a reference, and use it as a

running example to illustrate the problematic tackled in this paper and the contribution we made. According to the definition given by this case study, a CMS is “a system that facilitates coordination of activities and information flow between all stakeholders and parties that need to work together to handle a crisis”. Many types of crisis can be handled by such systems, including terrorist attacks, epidemics, accidents. To illustrate the case study, they provide an instance of a CMS in the context of car accidents. They define this system as the following:

“The Car Crash CMS (CCCMS) includes all the functionalities of general crisis management systems, and some additional features specific to car crashes such as facilitating the rescuing of victims at the crisis scene and the use of tow trucks to remove damaged vehicles.”

The requirement document defines ten use cases, described using textual scenario. Each scenario defines first a *main success scenario* which represents the normal flow of actions to handle a crisis (*e.g.*, retrieve witness identity, contact firemen located near to the crash location). Then, a set of *extensions* are described to bypass the normal flow when specific actions occurs (*e.g.*, witness provides fake identification, firemen are not available for a quick intervention).

2.2 Composition framework: ADORE

The ADORE framework defines a compositional approach to support complex business processes modeling, using the orchestration of services paradigm. Models describing business-driven processes (abbreviated as *orchestrations*, defined as a set of partially ordered activities) are composed with process fragments (defined using the same formalism) to produce a larger process. *Fragments* realize models of small behavior and describe different aspects of a complex business process. ADORE thus allows a business expert to model these concerns separately and then compose them.

Using ADORE, designers can define *composition units* (abbreviated as *composition*) to describe the way fragments should be composed with orchestrations. The merge algorithm used to support the composition mechanism [?] computes the set of actions to be performed on the orchestration to automatically produce the composed process. Implementation details, environment screenshots and video demonstrations are available on the project web site¹.

We proposed in our previous work [?] a realization of the CCCMS system using ADORE. We realized all the use cases main scenarios as orchestration of services, and extensions as fragments to be integrated into these orchestrations. The complete set of designed models (12 orchestrations & 24 fragments, representing 196 activities scheduled by 224 relations in terms of implementation) is available on the CCCMS realization web page²

¹ <http://www.adore-design.org>

² <http://www.adore-design.org/doku/examples/cccms/>