Model-Driven Engineering of Composite Service Oriented Applications

Bill Karakostas, City University London, UK
Yannis Zorgios, CLMS Ltd., UK

ABSTRACT

Composite applications integrate web services with other business applications and components to implement business processes. Model-driven approaches tackle the complexity of composite applications caused by domain and technology heterogeneity and integration requirements. The method and framework described in this paper generate all artefacts (workflow, data, user interfaces, etc.), required for a composite application from high level service oriented descriptions of the composite application, using model transformation and code generation techniques.

Keywords: Business Objects, Business Process Modelling, Composite Application Frameworks, IDEF0, Model Driven Service Engineering, Service Composition

INTRODUCTION

Contemporary application development is faced with multiple challenges, caused mainly by the new turbulent business environment in which organisations (and consequently their IT systems) operate. Agility, flexibility, resilience, adaptability and Web enablement, are no longer options but mandatory requirements that need to be engineered in today’s applications and systems.

For almost a decade now, Model Driven Engineering (MDE) under various initiatives such as MDA (Miller & Mukerji, 2001), has promised to transform the speed at which applications are developed. By advocating model-based system specification and code generation via model transformations techniques, to replace traditional specify-design-code system lifecycles, MDE promises enhanced quality resulting from less manual coding and thus less scope for programming errors.

Service Oriented Architecture (SOA) plays an orthogonal role to MDE, by unifying concepts and approaches to software development under the single paradigm of a service. The natural synergy and complimentarity of MDE and SOA, has resulted in the new paradigm of model-driven service engineering (Marcos et al., 2009). This new paradigm advocates the use of service models across all phases of software development, with service modelling techniques.

DOI: 10.4018/jitsa.2011010102
replacing the manual crafting of software (web) services. SOA helps to align IT capabilities to achieve business goals in offering value propositions, for instance to “orchestrate” lower level IT infrastructure services (Chen et al., 2009).

To date, model driven service engineering has been primarily applied to the development of systems composed purely of services, e.g. to automate tasks of service composition (Orriens et al., 2003) or orchestration (Mayer et al., 2008). Typical business applications, however, are a combination of automated tasks that can be performed by service or other application components, and of manual (user driven activities), all underpinned by technologies such as middleware, DBMS, Web, etc. Thus, significant productivity improvements can apply only by tackling the whole spectrum of application development, not only the service part.

The aim of Model-Driven SOA is to create Service-Oriented Business Applications (SOBA) that truly support an organization (den Haan, 2009). Composite service oriented applications therefore simplify integration by allowing access and integration of heterogeneous information resources and automation of business processes, using SOA principles and architectures.

Service oriented business applications will need to integrate underlying technologies for process and data management, workflow, user interface etc. Service oriented models of business processes need to directly lead to executable code, either by direct execution (e.g. using a workflow engine, rule engine etc), or by transformations to executable models. Thus, as we argue in this paper, model driven service oriented application development is well suited to the complex business and integration requirements of contemporary composite applications. This paper proposes a method and framework for achieving model driven engineering of composite service oriented applications.

The following section of the paper explains the features of composite applications. Then, the paper reviews existing composite application frameworks and introduces a model driven service oriented composite application development methodology. It explains the steps of the method as well as the underlying foundations (meta-models), using the issue management business process as a case study. Finally, the paper compared the discussed approach with related service oriented engineering methodologies and techniques, and identifies directions for further research.

FEATURES OF COMPOSITE SERVICE ORIENTED APPLICATIONS

A composite application integrates functionality provided by different sources such as conventional applications, systems such as middleware, DBMS, or web services.

Composite applications need therefore to introduce an orchestrating framework to coordinate the interactions between their constituting components, to produce the new, derived, functionality.

Multiple web services combined in composite applications require interoperable mechanisms to control business process activities start, end, success, failure etc, to create, access and manage process context information, and to inform participants of changes to an activity. Composite applications might also need to work with several transactional models such as one and two phase transactions, long running transactions and so on (OASIS, 2003).

Environments for building service oriented composite applications need to facilitate easy and visually composition of the business services in a way that makes sense to IT or business analysts (SAP, 2005). Ensuring such business services are reusable across the enterprise via some form of repository such as UDDI is also critical. Finally, providing the ability to assemble these different business services together to form the actual composite application, without coding, is also critical (Frye, 2005).
Related Content

Seeds of Knowledge: Nuggets, Memes, and the Search for the Basic Units
[www.igi-global.com/chapter/seeds-knowledge-nuggets-memes-search/24868?camid=4v1a](www.igi-global.com/chapter/seeds-knowledge-nuggets-memes-search/24868?camid=4v1a)

Introduction
Brian Lehaney, Steve Clarke, Elayne Coakes and Gillian Jack (2004). *Beyond Knowledge Management* (pp. 1-11).
[www.igi-global.com/chapter/introduction/5548?camid=4v1a](www.igi-global.com/chapter/introduction/5548?camid=4v1a)

Performance Implications of the Knowledge Chain
[www.igi-global.com/article/performance-implications-knowledge-chain/2669?camid=4v1a](www.igi-global.com/article/performance-implications-knowledge-chain/2669?camid=4v1a)
Integrating IS Security with Knowledge Management: Are We Doing Enough?
[www.igi-global.com/article/integrating-is-security-with-knowledge-management/117901?camid=4v1a](www.igi-global.com/article/integrating-is-security-with-knowledge-management/117901?camid=4v1a)