

Chapter 9

Characterising Enterprise Application Integration Solutions as Discrete- Event Systems

Sandro Sawicki
UNIJUI University, Brazil

Fabricia Roos-Frantz
UNIJUI University, Brazil

Rafael Z. Frantz
UNIJUI University, Brazil

Iryna Yevseyeva
Newcastle University, UK

Vitor Manuel Basto Fernandes
Polytechnic Institute of Leiria, Portugal

Rafael Corchuelo
University of Seville, Spain

ABSTRACT

It is not difficult to find an enterprise which has a software ecosystem composed of applications that were built using different technologies, data models, operating systems, and most often were not designed to exchange data and share functionalities. Enterprise Application Integration provides methodologies and tools to design and implement integration solutions. The state-of-the-art integration technologies provide a domain-specific language that enables the design of conceptual models for integration solutions. The analysis of integration solutions to predict their behaviour and find possible performance bottlenecks is an important activity that contributes to increase the quality of the delivered solutions, however, software engineers follow a costly, risky, and time-consuming approach. Integration solutions shall be understood as a discrete-event system. This chapter introduces a new approach based on simulation to take advantage of well-established techniques and tools for discrete-event simulation, cutting down cost, risk, and time to deliver better integration solutions.

DOI: 10.4018/978-1-4666-8823-0.ch009

INTRODUCTION

Frequently the current business processes in an enterprise has to be improved or new ones have to be created to support the enterprise's daily activities. All over the years, enterprises have been purchasing off-the-shelf software as well as developing custom applications aiming at giving computational support to implement and run their business processes. As a result it is not difficult to find an enterprise which has a software ecosystem composed of applications that were built using different technologies, data models, operating systems, and most often were not designed to exchange data and share functionalities. It is common that, from an administrative point of view, such software ecosystems are viewed as homogeneous systems, and are expected to support the enterprises' business processes by using current data and functionality without requiring too much investment. Enterprise Application Integration (EAI) provides methodologies and tools to design and implement integration solutions. The goal of an integration solution is to keep a number of applications' data in synchrony or to develop new functionality on top of them, so that applications do not have to be changed and are not disturbed by the integration solution (Hohpe & Woolf, 2003).

Amongst the state-of-the-art integration technologies available for enterprises to design and implement integration solutions, there are Camel (Ibsen & Anstey, 2010), Spring Integration (Fisher, Partner, Bogoevici, & Fuld, 2010), Mule (Dossot & D'Emic, 2009), and Guaraná (Frantz & Corchuelo, 2012). These technologies provide a domain-specific language that enables the design of conceptual models for integration solutions. Such languages follow the Pipes and Filters architectural style and provide support for using integration patterns (Hohpe & Woolf, 2003). In the Pipes and Filters style, a larger process is divided into a number of smaller and independent services (Filters), which are usually desynchronised by channels (Pipes). The integration patterns document a set of well-known services that are directly related to integration tasks.

The analysis of enterprise application integration solutions to predict their behaviour and find possible performance bottleneck is an important activity that contributes to increasing the quality of the delivered solutions. It is not enough to design a conceptual model for an integration solution, but it is also essential to analyse its behaviour under different workloads and minimise performance bottlenecks. Most often, the approach adopted by software engineers requires the construction of the integration solution, the execution of the actual integration solution, and the collection of data from this execution. This is a costly, risky, and time-consuming approach. The construction of integration solutions is expensive because it demands a good command on an integration technology to map the conceptual model into executable code. The execution involves the setup of a controlled environment in which the integration solution can be deployed, the generation and injection of input data into the integration solution, and the emulation of critical running scenarios. Any faults in the constructed solution may cause the execution to fail and negatively affect the analysis. The collection of data requires the insertion of extra code into the constructed integration solution and the configuration of the runtime system of the adopted integration technology that enables monitoring and collecting. A new approach that enables the analysis of the behaviour and discovering possible performance bottlenecks still in the design phase, taking as input the conceptual models of application integration solutions, would help to reduce cost, risk, and time spent.

Simulation is a research field that deals with experimentation of models to make predictions about the behaviour and the performance of actual systems. This chapter focuses on systems whose models are classified as stochastic, dynamic, and discrete. Stochastic models use probability to model actual systems in which uncertainty is present, which makes them a good option to represent the real-world. Integration

26 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the product's webpage:
www.igi-global.com/chapter/characterising-enterprise-application-integration-solutions-as-discrete-event-systems/137443?camid=4v1

This title is available in *Advances in Systems Analysis, Software Engineering, and High Performance Computing*, InfoSci-Books, InfoSci-Engineering, Science, Engineering, and Information Technology. Recommend this product to your librarian:

www.igi-global.com/e-resources/library-recommendation/?id=107

Related Content

Abstraction and Implementation Strategies for Augmented Reality Authoring

Florian Ledermann, Istvan Barakonyi and Dieter Schmalstieg (2007). *Emerging Technologies of Augmented Reality: Interfaces and Design* (pp. 138-159).

www.igi-global.com/chapter/abstraction-implementation-strategies-augmented-reality/10162?camid=4v1a

Exploring Emergence within Social Systems with Agent Based Models

Marcia R. Friesen, Richard Gordon and Robert D. McLeod (2014). *Interdisciplinary Applications of Agent-Based Social Simulation and Modeling* (pp. 52-71).

www.igi-global.com/chapter/exploring-emergence-within-social-systems-with-agent-based-models/106761?camid=4v1a

Simulation and Modelling Knowledge-Mining Architectures Using Recurrent Hybrid Nets

David Al-Dabass (2008). *Simulation and Modeling: Current Technologies and Applications* (pp. 266-306).

www.igi-global.com/chapter/simulation-modelling-knowledge-mining-architectures/28990?camid=4v1a

Virtual Modelling of Prehistoric Sites and Artefacts by Automatic Point-Cloud Surveys

Mercedes Farjas, Francisco J. García-Lázaro, Julio Zancajo, Teresa Mostaza and Nieves Quesada (2011). *Virtual Technologies for Business and Industrial Applications: Innovative and Synergistic Approaches* (pp. 201-217).

www.igi-global.com/chapter/virtual-modelling-prehistoric-sites-artefacts/43412?camid=4v1a