

## Chapter 3

# A Cross-Platform Architecture with Intelligent Agents for Dynamic Processes and Services Composition

Chung-Yeung Pang  
Sevecu AG, Switzerland

### ABSTRACT

*To meet business demands, enterprise software systems are required to be more dynamic, flexible and adaptive. Business processes must often be context-aware. Things get complicated when enterprise software systems, after a decade of evolution, comprise heterogeneous platforms and different technological stacks. This chapter presents the design and implementation of a cross-platform architecture with intelligent agents for dynamic business rules, process flows and services composition. The architecture includes an Enterprise Service Bus for service integration. Service agents are used to handle services. A Central Intelligent Agent that contains a Prolog-style rule-based engine is designed to execute business rules and processes. These agents are implemented in both Java and COBOL. Business process flows are completely rule- and context-driven. The services and components for the business processes are dynamically constructed. The proposed architecture and programming model enables fast prototyping and rapid development in an agile development process across different platforms.*

### INTRODUCTION

Enterprise IT systems have typically undergone a long period of evolution since the 1960s. These decade-long projects have eventually produced some of the most complex software systems in existence. For large corporations, mainframe applications programmed in COBOL often form the backbone of their IT structures.

By the mid-1980s, the object-oriented programming paradigm had evolved and new programming languages, such as C++, Smalltalk and, later on, Java, were introduced. In the meantime, multi-tier and

DOI: 10.4018/978-1-4666-8510-9.ch003

client server architectures had become popular. Many corporations used these new architectures for their new lines of business applications. The front-end applications were programmed in OO languages while the backend applications were still in legacy languages, like COBOL. Applications for these new lines of business often formed silos in the whole enterprise IT.

With the advances in Web and mobile technology, corporations must consider integrating Web-based and mobile applications with their existing applications. Furthermore, applications of different lines of business cannot generally be isolated. Service-oriented architectures (SOAs) have become very popular for application integration in recent years. Enterprise IT systems consist of different systems, applications and architectures of different ages, as well as technologies that need to interoperate both within and across organizations. This heterogeneity in enterprise IT systems has become a big problem for many corporations - the operation and maintenance costs for these systems are usually very high and, worst of all, they lack business agility.

Contemporary businesses have placed heavy requirements on enterprise IT infrastructure to operate in highly stable yet flexible and fast environments, as well as to introduce new features and operation processes in order to meet their continual growth. Despite their obsolescence, legacy systems continue to provide a competitive advantage by supporting unique business processes and by storing invaluable knowledge and historical data. New features and applications can no longer be developed in an isolated manner. They must leverage the existing legacy software components within a heterogeneous environment. This is quite a challenge for software development.

Software systems are required to become more dynamic, flexible and adaptive. Business processes must often be context-aware. For example, in European banks, new rules have been imposed on transaction-processing for customers from the USA. The new regulations of Basel III have also resulted in deviations from standard business processes for transactions that involve deposits over a certain period of time. Traditional software engineering methodologies are inadequate in addressing the complexity and dynamic nature of these business requirements. On the other hand, there are serious consequences for companies when such requirements are not implemented responsively. In fact, some regional banks within Europe have ceased doing business with US customers because their systems cannot cope with the changes required.

In this chapter, I present a solution to the challenges mentioned above. The solution involves the design and implementation of a cross-platform architecture with intelligent agents for dynamic business rules, process flows and services composition. An agile approach to application development in a heterogeneous enterprise's IT system is also presented. Finally, the migration of legacy components to modern systems is also addressed.

## **BACKGROUND**

### **Architecture, Programming Model, and Agile Development Process**

Advances in software technologies have shown a lot of promise. Back in the late-1980s, software scientists claimed that OO languages were for programming in the large (Wegner, 1989). The OO paradigm, with polymorphism and inheritance, was a solution to resolve and control the complexity of enterprise applications. In practice, the OO approach has often been found inadequate in managing the complexity of real-world applications. Take, for example, the modeling of US customer objects - we can model a

29 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/a-cross-platform-architecture-with-intelligent-agents-for-dynamic-processes-and-services-composition/135222](http://www.igi-global.com/chapter/a-cross-platform-architecture-with-intelligent-agents-for-dynamic-processes-and-services-composition/135222)

## Related Content

---

### Cooperation Between Agents to Evolve Complete Programs

Ricardo Aler, David Camacho and Alfredo Moscardini (2003). *Intelligent Agent Software Engineering* (pp. 213-228).

[www.irma-international.org/chapter/cooperation-between-agents-evolve-complete/24151](http://www.irma-international.org/chapter/cooperation-between-agents-evolve-complete/24151)

### Graphical Modeling of Security Goals and Software Vulnerabilities

David Byers and Nahid Shahmehri (2015). *Handbook of Research on Innovations in Systems and Software Engineering* (pp. 1-31).

[www.irma-international.org/chapter/graphical-modeling-of-security-goals-and-software-vulnerabilities/117918](http://www.irma-international.org/chapter/graphical-modeling-of-security-goals-and-software-vulnerabilities/117918)

### Lowering Coupling in Distributed Applications With Compliance and Conformance

José Carlos Martins Delgado (2020). *Applications and Approaches to Object-Oriented Software Design: Emerging Research and Opportunities* (pp. 35-77).

[www.irma-international.org/chapter/lowering-coupling-in-distributed-applications-with-compliance-and-conformance/249320](http://www.irma-international.org/chapter/lowering-coupling-in-distributed-applications-with-compliance-and-conformance/249320)

### Gamification: Model-Driven Engineering Approaches

Pedro Aguiar and Isabel Azevedo (2020). *Software Engineering for Agile Application Development* (pp. 141-163).

[www.irma-international.org/chapter/gamification/250440](http://www.irma-international.org/chapter/gamification/250440)

### An Improved Motion-Sensing Based Monitoring System for Smart Health Care

Ruiling Gao, Minghuan Zhao, Zhihui Qiu, Yingzhou Yu and C. Hwa Chang (2015). *International Journal of Software Innovation* (pp. 48-61).

[www.irma-international.org/article/an-improved-motion-sensing-based-monitoring-system-for-smart-health-care/133114](http://www.irma-international.org/article/an-improved-motion-sensing-based-monitoring-system-for-smart-health-care/133114)