Chapter 17
Legacy Software Integration in Service-Driven Environments:
An Intelligent Agent-Based Framework

Chung-Yeung Pang
Seveco AG, Switzerland

ABSTRACT

Maintaining and upgrading legacy systems is one of the challenges many enterprises face today. Despite their obsolescence, legacy systems continue to provide a competitive advantage by supporting unique business processes and acting as a repository for invaluable knowledge and historical data. However, enterprises would prefer to develop their applications with modern software technology instead of continuing to develop in the mainframe but leverage existing business processes and data from their legacy systems. This chapter presents an architectural framework and implementation methodology of a Central Intelligent Agent that is responsible for legacy integration. The framework uses an Enterprise Service Bus for service integration and agents to handle services. The Central Intelligent Agent uses a Prolog-style rule-based engine and context awareness for service handling and a complementary service agent on the mainframe side for legacy integration. The underlying framework provides a full set of functions to integrate legacy COBOL applications as services into the system without any programming effort in COBOL. The proposed technique enables fast prototyping and rapid development in an agile development process. It also facilitates legacy migration through successive and iterative processes.

DOI: 10.4018/978-1-4666-6178-3.ch017

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 the IT structure. The operation and maintenance costs of these applications are usually very high. Worst of all, is the lack of business agility. Introducing new features to meet the constant growth of a corporation takes far too long and costs too much.

Maintaining and upgrading *legacy systems* is one of the most difficult challenges that many companies face today. They struggle with the problem of modernizing these systems while keeping the day-to-day operations intact. Despite their obsolescence, legacy systems continue to provide a competitive advantage by supporting unique business processes and acting as a repository for invaluable knowledge and historical data.

Pang (2012) presented an approach to improve the business agility of legacy IT systems by modernizing COBOL application development. The special features of this approach included the introduction of a Service-oriented Architecture (SOA) for Web client integration, using a model-driven development approach with code generation following an agile development process. This development approach has been used successfully in many projects since 2004. Despite the success of these projects to produce high quality software, the approach addresses application development in the legacy mainframe platform. Many corporations would like to get away from further development in this platform. They prefer to develop their applications with modern software technology, but leverage existing business processes and data from their legacy systems.

Eventually, the legacy business processes and the database in the legacy systems will be replaced by newly developed software with successive processes and data migration. This development strategy leads to a number of technical challenges. First, we need an enterprise architecture that can incorporate software components running on different platforms. Second, we need a programming model that seamlessly allows the invocation of modules in different programming languages. Third, we must look at the performance issues of running the final applications. Computing resources in the mainframe are expensive. It takes a lot of resources to start a transaction in the mainframe and the number of transactions must be limited in any given application. This chapter presents an architectural framework and implementation methodology for legacy mainframe integration using the *Enterprise Service Bus* (ESB) and intelligent agents on the modern and legacy environments to coordinate the integration.

This chapter is organized as follows. The background section provides basic concepts and techniques that are prevalent in the industry to handle legacy integration in service-driven environments. The next section details the proposed architecture and implementation methodology that facilitates the integration of service clients with the mainframe legacy systems by detailing the service agents, enterprise service bus, scripting and rule engines, that enable the integration of different systems including Web services, mobile systems, and external systems in the Cloud with the mainframe host systems. A *Central Intelligent Agent* is described that is used to control