Chapter 1
Service-Driven Approaches to Software Architecture: Principles and Methodology

Raja Ramanathan
Independent Researcher, USA

ABSTRACT

Software Architecture has evolved from simple monolithic system designs to complex, multi-tiered, distributed, and componentized abstractions. Service-driven architectural approaches have been a major driver for enabling agile, cost-effective, flexible, and extensible software applications and integration solutions that support the business dynamics of today’s fast-paced enterprises. SOA and the SCA model have been the typical Service-driven architectural approaches used in enterprises today, to tackle the challenges of developing and implementing agile and loosely coupled software and enterprise integration solutions. Recent trends involve the use of Web APIs and RESTful architecture in the enterprise for agile service development and application integration. The goal of this chapter is to explore, discuss, and recommend methodologies for Service-driven Computing in the enterprise. Service versioning is detailed as a primary architectural approach for accommodating modifications to services during their life cycle. Service Mediation, Enterprise Service Bus, and Composition mechanisms including Enterprise Mashups are explored. The chapter also presents the business value of APIs in the enterprise and investigates the value-add to Social Media and Cloud enterprise initiatives. The typical phases of a Service-driven development life cycle are explained and service design patterns to facilitate the engineering of flexible service-based applications are described. The chapter concludes with thoughts on future opportunities and challenges in the area of Service-driven computing.
INTRODUCTION

Organizational agility and business process adaptability are key considerations for businesses looking to establish a competitive advantage in the marketplace. The ability of an organization to adapt itself very quickly to the changing needs of the business can make a positive impact on its growth and profitability.

Today’s businesses are complex, multi-faceted entities that transcend geographical boundaries, utilize a multitude of diverse technology platforms and implementation techniques, and deal with enormous amounts of coordinated data to make effective business decisions. Isolated data in “Silos” are not acceptable anymore. Businesses are working towards seamless integration of data and processes within the enterprise and with their external business partners in order to build agile and adaptable business models.

Software Services are autonomous, platform-independent, computational entities. They implement business functions that can be dynamically assembled into distributable and interoperable business services (Alonso et al., 2004), and invoked to execute the use cases implemented by the business functions. They are exposed to clients (consumers of the service) as a formal interface that can be published and discovered.

Service Oriented Architecture (SOA) is the typical Service-driven architectural approach used in enterprises today to tackle the challenges of software development and enterprise integration, allowing application developers to overcome many distributed enterprise computing challenges (Georgakopoulos & Papazoglou, 2008). The promise of Service-oriented Computing is a world of cooperating services, where application components are assembled with little effort into a network of services that can be loosely coupled to create flexible and dynamic business processes and agile applications that may span organizations and computing platforms (Papazoglou, Traverso, Dustdar, & Leymann, 2008). REST (REpresentational State Transfer) (Fielding, 2000) is another popular architectural style that is gaining momentum in the realm of service architecture and focuses on development agility and simplicity.

A Service-driven approach primarily uses SOA and RESTful architectures to accomplish its goal of facilitating the development and implementation of enterprise applications and integration solutions that support the dynamics of the business. A well designed, standards-based, Service-driven methodology can provision a flexible infrastructure and processing environment for the enterprise and provide a robust and secure foundation for leveraging services for enterprise integration of data and business processes. Efficiencies in the design, implementation, and operation of service based systems will enable organizations to align their information technology with their agile business practices.

Services are generally built independently of the context in which they are used, thus enabling Loose Coupling between the service consumer and the service provider. Service providers can perform varied functions that range from simple queries to complex business processes that involve sophisticated message transformations, dynamic routing of messages to other services, orchestration (discussed later) among multiple application services, and choreography (discussed later) among multiple layers of service providers.

Web Services (Jini Service and UPnP Service are other examples of service technologies) are currently the most widely used technology for implementing Service-driven applications; and they provide the basis for the development of business processes and enterprise integration mechanisms, using components and services that are distributed across the network and accessible through standard interfaces and protocols. The idea is to promote a building blocks approach to developing business services with minimal effort by assembling application components into ser-