ABSTRACT

Architects are faced with the problem of building enterprise scale information systems, with streamlined, automated internal business processes and web-enabled business functions, all across multiple legacy applications. The underlying architectures for such systems are embodied in a range of diverse products known as Enterprise Application Integration (EAI) technologies. In this tutorial, we highlight some of the major problems, approaches and issues in designing EAI architectures and selecting appropriate supporting technology. An architect’s perspective on designing large-scale integrated applications is taken, and we discuss requirements elicitation, architecture patterns, EAI technology and features, and risk mitigation. J2EE and .NET technologies are used to illustrate the capabilities of state-of-the-art integration technologies.

Categories and Subject Descriptors
D.2.11 [Software Architectures]: Data abstraction; patterns;
D.2.12 [Interoperability] Distributed objects

General Terms
Performance, Design, Economics, Reliability, Security

Keywords
Enterprise application integration, COTS, integration architectures

1. TUTORIAL OVERVIEW

Enterprise application integration (EAI) is concerned with the ease with which a software component can be usefully incorporated in to a broader application context. The value and lifetime of an application or component can frequently be greatly increased if its functionality or data can be used in ways that the designer did not originally anticipate.

Enterprise applications integrate multiple business systems that were not intended to work together. Integrating such systems is hard for many reasons. These include the heterogeneity of the platforms and programming languages, the diversity and complexity of each individual business system, and the difficulty of understanding the requirements for the resulting integrated applications.

Software architects undertake a number of crucial tasks during the design of integrated enterprise applications [1,2]. Amongst these are:

1. Helping understand the functional and non-functional requirements for the integrated applications
2. Creating the initial architectural blueprint for the integrated applications
3. Selecting suitable integration technologies that can fulfill the application requirements
4. Validating that the combination of the architecture and the integration technology used to build the enterprise-wide application are likely to be successful before a major implementation investment is made.

These tasks require an extraordinary range of skills of knowledge. Architects must understand business requirements and their technology strategies. They must understand and document the existing applications in the enterprise, and devise an integration architecture that supports the business requirements. They must have a deep understanding of specialized integration technology, the strengths and weaknesses of various platforms, and appreciate the influences and risks of emerging new technologies and standards. They must also be able to justify return-on-investments, and design integrated applications that can be built on time and within budget.

This tutorial aims to help architects and software engineers successfully carry out the above tasks. It does this by leveraging our experience in architect roles in major integration projects over the last decade. Specifically, it provides:

1. Approaches to ensure that all common integration requirements are considered. Following these approaches reduces the risks of important project requirements being overlooked.
2. A description of common integration architectures. Nearly all integrated applications exploit one or more of these architectures, and we explain the strengths and weaknesses of each.
3. A description of the general classes of integration technologies that are used to build integrated systems. These include messaging, message brokers and
transformation engines, adapters, business process orchestration and web services. We describe the features you can expect in each, when and why to use them, and their inherent strengths and weaknesses.

4. A collection of technical requirements for selecting an integration platform. This contains comprehensive and detailed functional and non-functional requirements for the range of useful features that integration technologies provide. By using these requirements, you’ll be able to rapidly construct the detailed technical requirements for an integration platform, and use this to reliably and consistently evaluate the various vendor offerings.

5. Technical descriptions of the various integration technologies that comprise the Java 2 Enterprise Edition (J2EE) and .NET platforms. These are the two predominant integration platforms in the market. We describe the strengths and weaknesses of these technologies, and how they fulfill the various requirements in the knowledge base.

The tutorial is structured as follows.

The first section explains why designing architectures for integrated enterprise applications is difficult, and explains the many inherent complexities that are project and technology related. It then explains the basic architectures that are used, including file transfer, messaging and service-based. Finally there’s a general description of integration technologies and their most important features. Throughout this section, we illustrate problems, architectures and technologies with a collection of simple examples of integration scenarios. This section provides the introductory knowledge for what follows.

The second section introduces a case study that we use throughout the remainder of the tutorial to illustrate architectures and technologies. In the context of this case study, we present the checklists that can be used to understand the requirements for the integrated applications. We then present the structure of the integration technology knowledge base and a range of detailed technical requirements from the knowledge base. These are related to the key integration technologies that we will use in the case study.

The third section introduces the J2EE and .NET technologies that are used for integration projects. These include the Java Messaging Service, MSMQ, BizTalk, the Java Connector Architecture, adapters for mainframe access, web services and XML support, and Enterprise System Buses. We explain how each of these technologies has various levels of support for the relevant non-requirements in the knowledge base. This highlights the strengths and weaknesses of each, and helps guide architects on when (and importantly when not) to use each in a solution. We then illustrate the use of these technologies to build solutions for the case study, and discuss strategies that architects can employ to provide early validation that their design and technology selections are likely to succeed.

Finally, we focus briefly on the emerging directions of enterprise integration architectures and technologies. Web Services are having a profound influence in integration, and as more and more web services specifications such as BPEL4WS become reality, this is likely to increase. The Java Business Integration specification is also currently in its final stages. This essentially adds business integration APIs to J2EE, and should promote standardization across J2EE implementations.

2. REFERENCES
