Chapter VI

Reengineering Legacy Systems Towards Web Environments

Ying Zou, Queen's University, Canada
Kostas Kontogiannis, University of Waterloo, Canada

ABSTRACT

With the widespread use of the Web, distributed object technologies have been widely adopted to construct network-centric architectures, using XML, Web Services, CORBA, and DCOM. Organizations would like to take advantage of the Web in its various forms of Internet, Intranet and Extranets. This requires organizations to port and integrate their legacy assets to distributed Web-enabled environments, so that the functionality of existing legacy systems can be leveraged without having to rebuild these systems. In this chapter, we provide techniques to re-engineer standalone legacy systems into Web-enabled environments. Specifically, we aim for a framework that allows for the identification of reusable business logic entities in large legacy systems in the form of major legacy components, the migration of these procedural components to an object-oriented design, the specification of interfaces of these identified components, the automatic generation of CORBA wrappers to enable remote access, and finally, the seamless interoperation with Web services via HTTP based on the SOAP messaging mechanism.

INTRODUCTION

With the widespread use of the Internet and pervasive computing technologies, distributed object technologies have been widely adopted to construct network-centric architectures, using Web services, CORBA, and DCOM. Organizations would like to take
advantage of the Web in its various forms of Internet, Intranet and Extranets. This requires organizations to migrate and integrate their legacy assets within distributed Web-enabled environments. In this way, the functionality of existing legacy systems can be leveraged without having to rebuild these systems.

However, legacy systems are not designed as Web applications. Generally, a Web application is a loosely coupled, component-based program, which consists of three tiers including user interface, business logic and related databases with a clear distinction. The front-end user interface is accessed through Web browsers. The business logic runs in a Web server and application server. The databases reside on back-end servers. Each component communicates with others through standard protocols such as HTTP and XML. In contrast, legacy systems are usually standalone systems with complicated structures with all three tires intermixed and tightly coupled. According to the decomposability of a legacy system, the architecture of a system can be decomposable, semi-decomposable, or non-decomposable (Brodie & Stonebaker, 1995). In a decomposable system, the user interface, business logic and the related databases can be considered as distinct components with well-defined interfaces. Such systems are the best candidates for the migration. In a semi-decomposable system, only interface and business logic are separate modules. The business logic and databases are not separable, due to their complex structure. Such systems are more difficult to migrate, as the interaction of the business logic and the databases is not easy to understand. In a non-decomposable system, the system appears as a single unstructured and monolithic module. Such systems are the most difficult to migrate, as they are treated as black boxes, and their functionality and design cannot be recovered.

Problem Definition

To modernize legacy systems with Web technologies, there are three major issues to be addressed, namely, Web accessibility, componentization, and platform transformation.

- Web accessibility is a crucial requirement for a Web application. In this respect, the legacy interface should be replaced with a Web-enabled interface by placing a wrapper on top of the legacy system, or by re-generating a new interface in HTML form.
- Componentization focuses on the extraction of reusable components from the legacy systems. These components contain the evolving business logic and functionality of the legacy system. Once the components are identified, they can be easily maintained, and their interface can be described using standards such as WSDL (Web Service Description Language) (WSDL). Furthermore, components can be integrated with other components to produce new business functional units.
- Platform transformation involves migrating monolithic legacy systems into modern platforms such as the Web. In this context, there are two major types of strategies to be taken in order to leverage legacy assets into Web-enabled environments, namely:
  - Migration of legacy system into Web-enabled environments, and
  - Integration of the legacy system with new applications.
Toward an Ontology of ICT Management: Integration of Organizational Theories and ICT Core Constructs
[www.igi-global.com/chapter/toward-ontology-ict-management/19449?camid=4v1a](www.igi-global.com/chapter/toward-ontology-ict-management/19449?camid=4v1a)

Application Service Provision: A Working Tool for Inter-Organizational Systems in the Internet Age
[www.igi-global.com/chapter/application-service-provision/24489?camid=4v1a](www.igi-global.com/chapter/application-service-provision/24489?camid=4v1a)