ABSTRACT

The banking industry has undergone a major change in recent years. Global competition has forced the industry to be more agile and customer focused in all its services. Banks can no longer function in isolation, but have to operate cutting across physical boundaries. Interoperability, scalability, maintainability, and security are the upcoming challenges for the banking industry. This has enthused software architects to develop suitable software development paradigms that can seamlessly integrate business functions across organizational boundaries. This chapter envisages a hybrid approach that uses the service-oriented paradigm along with the software agent technology as a possible solution to the growing issues of inter-and intra-bank operations.

INTRODUCTION

Evolution in information and communication technology (ICT) has radically influenced the business world—how transactions are being carried out, how information is being exchanged, and how business collaborations are being handled. The banking sector is no exception as it plays a vital role in facilitating all kinds of business-related financial transactions. Thus, there is a growing need for the banking sector to keep pace with the emerging requirements of the business sector by adopting appropriate technology for its effectiveness. The emphasis today is on providing banking services anywhere, anytime, to anybody, with the sole objective of enhancing customer outreach and flexibility in transactions. Towards this end, the adoption of Internet banking, mobile
banking, core banking, and a range of financial services through different delivery channels holds great promises to bring about a paradigm shift in the banking industry. However, the success of electronic banking system initiatives will largely depend on their effective deployment, interoperability, and automated transaction management. Besides these technical issues, certain basic business needs such as lowering costs, reducing cycle times, integration across banks, greater return on investment, creating an adaptive, and a responsive business model are of paramount importance. The fundamental problem is the lack of a consistent architectural framework within which applications can be rapidly developed, integrated, and reused. Thus, there is a need for a well thought of architectural framework that can facilitate the assembly of components and services for rapid and dynamic delivery of complex services. This should not only cater to the present requirements but must scale up to the needs of next-generation banking systems.

This chapter proposes a hybrid model for engineering banking applications, which are essentially large scale, distributed, and heterogeneous in nature. The model is an integration of service-oriented architectural concepts and the software agent technology. Both the components address two important aspects of building large-scale open systems. The notion of service provides a higher level of abstraction for organizing applications in large-scale, open environments. This view of service orientation can provide the basic conceptual building blocks for integrating heterogeneous applications. On the other hand, software agent technology addresses the requirements of autonomy, and reactive and proactive behavior of applications, which are key ingredients of an agile application environment.

BACKGROUND

In order to maintain global competitiveness, the banks today keep introducing innovative financial services for their customers in the form of basic banking services, asset management, investment banking, and a range of online services. Such financial services are accessible through a variety of channels and software applications. Branch office automation, automated teller machines (ATMs), call centers, Internet, e-mail, fax, and mobile devices are some of the means through which it is possible to access any banking service, at any time, in the most convenient manner. But the key challenge here is to develop each application independently and still be able to integrate them whenever necessary. This aspect of banking application development can be appropriately addressed by adopting the service-oriented architecture (Shan, 2004).

Service-Oriented Architecture (SOA)

For a long time, systems were built following monolithic architectures that are highly fragile, customer specific, with non-reusable applications that are tightly coupled. But today there is a shift in the paradigm wherein software systems are built around a loosely coupled principle. There is an evolution in the software design paradigm where software systems are viewed as a collection of interacting service components (Kruger & Mathew, 2004). A service, in this context, refers to a set of functions provided by a software system that is accessible by an application program. In the rest of the text, the word function is used to refer to low-level operations, whereas the word service has been used as a higher-level concept. Service-oriented computing (SOC) is a computing paradigm that utilizes services as fundamental
Related Content

Payment Systems in Japan
[www.igi-global.com/chapter/payment-systems-japan/54704?camid=4v1a](www.igi-global.com/chapter/payment-systems-japan/54704?camid=4v1a)

The Relationship Between Institutional Investment and Earnings Management: Empirical Evidence from Turkey
[www.igi-global.com/article/the-relationship-between-institutional-investment-and-earnings-management/107002?camid=4v1a](www.igi-global.com/article/the-relationship-between-institutional-investment-and-earnings-management/107002?camid=4v1a)

The Impact of Ownership Structure on Firm Performance: Evidence from Pakistan
[www.igi-global.com/article/the-impact-of-ownership-structure-on-firm-performance/164986?camid=4v1a](www.igi-global.com/article/the-impact-of-ownership-structure-on-firm-performance/164986?camid=4v1a)

[www.igi-global.com/article/a-study-on-factors-influencing-mutual-fund-portfolio-performance/212734?camid=4v1a](www.igi-global.com/article/a-study-on-factors-influencing-mutual-fund-portfolio-performance/212734?camid=4v1a)