The emerging field of Web Services (WS) enables different software components to be integrated without having to develop them from scratch and without the hassle of custom coding. It is significantly changing the way in which applications and supporting infrastructures are integrated. These changes have led to the design and development of service-oriented architecture (SOA) for application systems. WS are becoming the basic building blocks using which new applications are created, and service composition is becoming the main focus of the application development process. Service composition combines two or more WS following a certain composition pattern to achieve a business process goal. Thus, service composition provides a mechanism for seamless integration of cross-enterprise and intra-enterprise applications.

The SOA is built on a foundation of Web standards, which define the roles and activities of the architectural elements, and thus support the interoperability of incompatible systems across the Web. The SOA is distributed, permitting elements of an application to be deployed on multiple systems and executed across networks. Because the transport mechanism is built on HTTP (Hyper Text Transfer Protocol), it is possible for application elements to interact within and across enterprises. The elements of an application are designed to support specific tasks within a broader workflow or business process. Each of these service elements of an application is responsible for defining its inputs and outputs using WS standards so that other elements are able to determine how this element operates, how to make use of its functionality, and what results to expect from its execution. Since WS technologies provide a language and environment neutral programming model, they facilitate application development that is evolutionary, building on IT investments previously made within an organization and developing new capabilities incrementally.

WS-based e-business systems are currently in the early phase of adoption, primarily within large organizations that have well-established IT infrastructures and technically savvy staff. Most of these organizations are experimenting with WS for application integration and developing innovative implementations within their e-business context. The deployment of WS is currently focused in two areas: (1) enterprise application integration - integrating disparate components (mainly legacy applications) of an enterprise wide system; and (2) business partner integration and secure exchange of data with dedicated partners without using a proprietary approach.
Despite the advantages of WS and their increasingly vital role in integration initiatives, first within an organization and then between enterprises, several challenges exist with developing and using them. A thorough and systematic investigation of these challenges is needed to help improve the rate of deployment of WS across organizations. WS computing poses significant challenges as developers determine how to leverage emerging technologies to automate individual applications based on cross-organizational, heterogeneous software components. WS standards permit application-to-application interoperability, but the coordination of a set of WS working toward a common end is still an open issue. The service providers need to learn how to develop WS interfaces that perform well, identify performance problems when they occur, and how to fix them.

In order to develop e-business applications using leased components through WS, the challenges from the perspective of the three stakeholders of WS, namely the service providers, service consumers and the WS Standards Organizations, have to be thoroughly understood. While some initial work has been reported in the literature, further research is needed to investigate the challenges that exist for each of the stakeholders, and develop a framework that organizes and inter-relates these challenges in an easily understandable manner. This will help improve our understanding of the factors that impact the deployment and use of WS. This framework can also be used to analyze the challenges in small and medium-sized enterprises (SMEs), as they play a vital role in generating employment opportunity and turnover within many major economies globally. Particularly, as large companies downsize and outsource more functions, the importance and role of SMEs in the economy is increasing worldwide.

Most of the problems related to WS based software development stem from WS adoption and use in organizations. The IS discipline has a long history of adoption and diffusion research and has developed a number of theories and frameworks to investigate such problems. These theories and frameworks should be drawn upon to test different propositions regarding organizational, and industry level use of WS. The framework for studying the challenges of WS adoption and deployment should include both technical and non-technical issues associated with WS. Also, these issues have to be investigated for each of the major stakeholders, namely, web service providers, web service consumers, and standards organizations.

ABOVE THIS ISSUE

In the lead article titled “Assisting Learners to Dynamically Adjust Learning Process through Software Agents,” Weidong Pan and Igor Hawrysiwycz discuss the use of intelligent agents for helping readers learn at their own pace in an online learning environment. They have developed an agent-based system that provides a variety of support services. It assists learners to develop personalized learning plans and guides them to dynamically adjust the learning process towards their goals. The online learning process is first discussed and the structure of the UOL (unit of learning) database that provides links between a practical learning scenario and the required services is presented. The multi-agent archi-
tecture is configured based on the roles of the agents involved. The algorithms used for guiding the learners to dynamically adjust their learning process are also described.

In the second article, Lei Song, Xining Li and Jingbo Ni discuss an approach for locating hosts that provide services requested by mobile agents. In their article titled “A Database Service Discovery Model for Mobile Agents,” they present a new model for solving the database service location problem in the mobile agent domain by implementing a Service Discovery Module based on search engine techniques. With a typical interface provided by a mobile agent server, their Service Discovery Module improves the decision ability of mobile agents with respect to information retrieval. They discuss the design of an independent search engine called IMAGOSearch that facilitates service discovery by intelligent mobile agents.

The third article titled “An Agent-Based Framework for Emergent Process Management,” by John Debenham discusses a framework for managing business processes using intelligent agents. The amount of process knowledge that is relevant to a knowledge-driven process can be enormous and may even include common sense knowledge.

In an e-market domain, majority of the transactions are knowledge-driven and multi-agent systems are useful for managing these complex business processes. Such systems use agents that are driven not by a process goal, but by an inflow of knowledge, where each chunk of knowledge may be uncertain. These agents assess the extent to which they should believe the information, and hence require an inference mechanism that can cope with differing integrity levels of information.

In the last article titled “Importance of Interface Agent Characteristics from End-User Perspective,” Alexander Serenko reports on his empirical investigation of user perceptions on the importance of interface agent characteristics. Interface agents are software entities that are incorporated into various computer applications including electronic mail systems. There are several characteristics of interface agents that require special attention from agent developers. In order to identify the significance of these characteristics, Serenko surveyed a group of actual users of an email interface agent. The results indicate that information accuracy and the degree of usefulness of an agent are the most salient factors, followed by user comfort, extent of user enjoyment, and visual attractiveness.