Chapter XLIX
Web Services and Service-Oriented Architectures

Bruce J. Neubauer
University of South Florida, USA

INTRODUCTION

A review of the development of information systems can help in understanding the potential significance of Web services and service-oriented architecture (SOA) in the public sector. SOA involves the convergent design of information systems and organizational workflows at the level of services. The purpose of this chapter is to suggest a strategy for mapping the design of service-oriented architectures onto the complex patterns of governance including combinations of federalism, regionalism, and the outsourcing of functions from government agencies to nonprofit organizations. This involves the modeling of workflows and the identification of opportunities for the sharing of services among agencies and nonprofits.

The structures of government agencies reflect political jurisdictions, legislative committee structures, areas of public policy, and geographical locations. Federalism creates situations in which multiple agencies (often at different levels of government) have similar responsibilities in the same geographic areas. Metropolitan areas are complex mosaics of local governments and special districts. In addition, nonprofit organizations are also involved in strategic alliances with government agencies to provide services to citizens. The coordination of efforts among multiple organizations has been one of the major functions of public administrators acting through formal or informal networks of relationships within and across organizational boundaries. Web services and SOA can be used to help integrate the often costly and fragmented delivery of government services.

BACKGROUND

Information systems were historically centralized. It is still common for departments within an organization to each have their own computer applications used to support accounting, payroll, or other specific responsibilities. Such applications and systems are sometimes referred to as "stovepipe" applications, suggesting the fact that they stand alone. While such applications may well optimize specific functions, they may not be

designed to support business process workflows that cut across departments. As a consequence, common business processes often require manually entering data multiple times and making adjustments for the different ways the same data are stored in multiple databases.

Modern systems of applications are sometimes called enterprise systems or ERPs. The acronym ERP stands for enterprise resource planning. Examples of ERP solutions include SAP and Oracle applications. Such systems are designed at the scale of an entire organization. ERPs are modular in nature. An organization can purchase the modules it needs from the same vender and be assured that they are compatible with one another. The benefit of systems of computer applications designed at the enterprise level is an improved ability of the applications to support entire business processes. Many enterprise systems share one enterprise database. It should not be necessary to enter data into multiple stovepipe applications. The underlying database should provide top managers with one version of the truth rather than a collection of summaries gleaned from multiple databases within the organization. Electronic data interchange (EDI) is a similar technology for creating data communications between organizations. EDI solutions tend to be point-to-point connections between specific organizations for specific purposes.

Government agencies can more easily coordinate their efforts and outsource responsibilities when they have modern information systems. ERP and EDI technologies are valuable and widely used. However, ERP solutions tend to be inflexible and EDI solutions may not support agile relationships among multiple organizations in networks of strategic alliances. Web services is a modern aspect of computer programming that facilitates the linking together of disparate legacy computer resources. SOA is an application of design principles to both computer resources and organizational units that can facilitate the creation of agile relationships between departments within large organizations and among organizations working together.

Early computer programmers worked in procedural programming languages such as FORTRAN and COBOL. The entire program usually ran on one machine and data were often managed by the software application itself rather than by the use of a separate database management system. Later, when programmers wrote procedural applications to run on networked computers, they sometimes used remote procedure calls to make use of subprocedures of code physically residing on another machine. The application would send a request across the network and wait for the subprocedure to run on the other machine and send data back to the application. This way of thinking evolved into object-oriented programming. A major benefit of this approach is reuse of the work required to create the subprocedure. Another benefit relates to loose coupling, meaning that the task of building a very large and complex application can be assigned to many individuals who can each build his or her own part of it without a detailed knowledge of what other programmers are doing (Kaye, 2003). As long as the interface between the part of the system and the rest of the system is agreed upon, how the part does what it does is not especially important to other programmers.

To an object-oriented programmer, a use of a Web service is the invocation (calling) of structured code residing across a network on a Web server. To a business analyst, a Web service is an opportunity to reuse not just code but entire services provided by organizations. Service-oriented architecture involves the orchestration or choreography of multiple services. An orchestration involves one focal application calling one or more services as needed within the same organization. A choreography is a coordination of services between or among organizations. Having been asked to make the transition from procedural programming to object-oriented programming, computer programmers are now facing the challenge of working at the level of the design of